B33

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
21 December 2000 (21.12.2000)

PCT

(10) International Publication Number
WO 00/77664 A2

(51) International Patent Classification^7: G06F 17/00

(21) International Application Number: PCT/US00/15938

(22) International Filing Date: 8 June 2000 (08.06.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/330,904 11 June 1999 (11.06.1999) US

(63) Related by continuation (CON) or continuation-in-part (CIP) to earlier application:
US 09/330,904 (CON)
Filed on 11 June 1999 (11.06.1999)

(71) Applicant *(for all designated States except US)*: LIBERATE TECHNOLOGIES LLC [US/US]; 2 Circle Star Way, San Carlos, CA 94070-6200 (US).

(71) Applicant *(for US only)*: BOUCHER, Antoine [CA/CA]; 32-270 North Centre Road, London, Ontario N6G 5E2 (CA).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: MCRAE, Paul, E. [CA/CA]; 141 Chesham Place, London, Ontario N6G 3T7 (CA). SCHEYEN, Peter, G., N. [CA/CA]; 20 Inverary Road, London, Ontario N6G 3L6 (CA).

(74) Agent: BRUCKNER, John, J.; Wilson Sonsini Goodrich & Rosati, 650 Page Mill Road, Palo Alto, CA 94304-1050 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *Without international search report and to be republished upon receipt of that report.*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: METHODS, APPARATUS, AND SYSTEMS FOR STORING, RETRIEVING AND PLAYING MULTIMEDIA DATA

(57) Abstract: Various embodiments of the invention provide increased speed and decreased computer processing for playing and navigating multimedia content by using two types of data objects for displaying the multimedia content. The data object type includes rendered multimedia content data. The second data object type provides semantic content corresponding to the rendered multimedia content. The storage medium in which these two types of data objects are contained is referred to as a rendered cache. The semantic content can include locations, sizes, shapes, and target universal resource identifiers of hyperlinks, multimedia element timing, and other content play instructions. The very fast play of content stored in the rendered cache is due to the elimination of the steps of laying out the content, rendering the content, and generating the semantic representation of the content. These steps are required each time the content is played after retrieval from a conventional cache. The only steps required for playing content from the rendered cache are to read the rendered content, read the semantic content, restore the semantic representation, and play the content. A traditional web browser visiting a web site that resides in a rendered cache provides an almost instantaneous display of the web site. The caching mechanism provided by various embodiments of the invention is independent of content file format and the stored semantic content file format. As long as a client application, such as a content browser, can recognize and play the multimedia content and recognize and interpret the semantic content, the application can realize the benefits provided by the rendered cache.

METHODS, APPARATUS, AND SYSTEMS FOR STORING, RETRIEVING AND PLAYING MULTIMEDIA DATA

BACKGROUND OF THE INVENTION

Field of the Invention

The invention relates generally to improvements in computer systems. More particularly, the invention relates to methods, apparatus, and systems storing multimedia content such as audio, text, image, and graphical content in a cache directory.

Discussion of the Related Art

Prior art graphics processing storage medium, sometimes called a cache system, is known to those skilled in the art. For example, a conventional caching system is typically composed of a small fast storage device that contains a "snapshot" of information originally received from a larger, slower source. The snapshot is considered by the particular implementation to be the most relevant information to the processing occurring during the current time period.

In the context of Internet content, a "cache" is a file, database, directory, or set of directories disposed in a computer file system. The cache stores content that has been previously retrieved generated or otherwise produced. Internet browsers and editors use cache directories to store content. The cached content is used in place of remote content whenever possible in order to decrease retrieval latencies. Therefore, many web browsers and text editors save Internet and other text and graphical content in a cache directory in order to reduce access times. This content is usually stored in its original form [for example, hypertext markup language (HTML) and accompanying images].

A problem with this technology has been that to view content based on the image data stored in the cache typically requires layout and rendering of the data. If the data upon which content is based does not change, the process of rendering need only occur once to a display buffer. When information is video must be marked with an appropriate active area for that car. Often, this is accomplished by image recognition. In other words, during an editing process, video editing apparatus recognizes if the car has moved from the right side of the video screen to the left side of the video screen. If this has happened, the editing apparatus will establish a new active area for that car on the left side of the screen for those video frames where the car is on the left side of the screen. Unfortunately, there are several problems with image recognition. For example, image recognition systems typically use the color of an object to track its location. If the object moves into an area where the lighting changes, e.g. because a shadow is cast on the object, that can prevent accurate image recognition. In addition, image recognition has other problems that can introduce errors into the process of marking an object.

Image recognition is well known in the art and is discussed by J.P. Lewis in "Fast Normalized Cross-Correlation", http://www.idiom.com/~zilla/Papers/nvisionInterface/nip.htm; U.S. Patent 5,267,332, issued to Walch et al.; U.S. Patent 4,611,347, issued to Netravali et al.; U.S. Patent 4,868,871, issued to Watson, III; R.O. Duda et al., "Pattern Classification and Scene Analysis", published by John Wiley and Sons in 1973; and R.C. Gonzalez et al., "Digital Image Processing", 3d edition, published by Addison-Wesley in 1992. Lewis, Walch, Netravali, Watson, Duda and Gonzalez are incorporated herein by reference.

Instead of using image recognition apparatus, a person could mark each frame of the video manually. However, marking each frame of the video would be very tedious and time-consuming. What is needed is a novel method for marking moving objects that reduces the amount of human interaction required.

Summary

A method in accordance with the invention comprises the step of annotating a video clip with a link. (The video clip can be animation, and as used herein, "video" includes animation.) This link can be a link to a web page or other source of information, or a link to application software. In one embodiment, a first set of frames within a video clip is annotated, e.g. using video editing apparatus. During this step of annotation, a portion of each annotated frame is designated as an "active area." This means that if one moves a cursor to the active area and clicks within the active area, a link associated with the active area will be executed.

The active area is typically associated with an image of an object, e.g. a car racing around a track, an actor in a scene, a product displayed in a commercial, a cartoon character, or some other person or thing. Accordingly, during annotation of a frame, a person annotating the video clip will establish the location of the active area associated with that object. (The active area is typically a rectangular or circular region surrounding the object in question.)

It would be extremely unwieldy to annotate every frame in a video clip in this manner. Accordingly, in accordance with the invention, only a subset of the frames (e.g. every tenth frame) is annotated. The location of the active area in the other frames is established by interpolation. The location can be established by a linear interpolation technique. Alternatively, the location can be established by an interpolation technique that generates a curve, such as a spline interpolation technique.

In another embodiment of a method in accordance with the invention, a subset of frames, for example two frames such as the first and last frames of a sequence of frames, are manually annotated. A video editing system then calculates the position of the active area for each frame between these manually annotated frames of the sequence based on an interpolation algorithm. The system then determines the difference between the interpolated position of the active area and the position of the object with which the active area is associated for each of the frames. This calculation is based on image recognition. The image recognition can be carried out by image recognition software. Alternatively, image recognition can be performed by a combination of hardware and software. (Image recognition can also be performed by hardware. The partitioning between hardware and software can be done in different ways.) Thereafter, the system highlights those frames that the system has determined to exhibit the greatest error. In other words, the system highlights those frames in which the distance between the active area and the object associated with that active area is greatest. One skilled in the art can define a distance between an active area and the object based on maximum, minimum or average distance between selected area and points within the object.

In one embodiment, the system flags the frame with the largest error. In another embodiment, the system flags the n frames with the n largest errors, where n is an integer. In another embodiment, the system flags all frames in which the error is over a certain threshold.

Thereafter, the person annotating the video clip can observe the flagged frames and decide whether he or she desires to manually annotate those frames. The person annotating the video clip also has the option of recalculating the interpolation. After recalculating the interpolation, the user can see if other frames are flagged that might require manual annotation.

Flagging may be accomplished by highlighting the flagged frame on a display or by providing some other form of output indicator.

In accordance with another aspect of the invention, a video signal is sent to one or more video viewing systems so that one or more viewers can watch the video signal. The video signal can be broadcast to many viewers. The location of one or more active areas associated with a video signal is also communicated to the one or more video viewing systems for some of the frames of the video signal, but not other frames of the video signal. The one or more video viewing systems determine the location of the one or more active areas for the other frames by interpolation. In this way, instead of having to provide data to the video viewing systems concerning the active area locations for each frame, such data is provided only for selected frames, and the video viewing systems calculate where the active areas for the other frames should be by interpolating, e.g. using either a linear interpolation technique or a spline interpolation technique. Because the source of the video signal need not provide active area information for each frame of the signal, the bandwidth requirements for communicating this information are reduced.

In one embodiment, not only is the location of the active area communicated in this way, but the size of the active area is communicated in this way as well.

In accordance with another aspect of the invention, a video signal comprising a set of frames is communicated along with the location of an active area for some of the frames but not others of the frames. The active area corresponds to an object depicted in those frames. The object can be a product, a person, a cartoon character, or anything else that can be depicted in a video. As mentioned above, the active area is not communicated for the others of the frames, although the object is depicted in the others of the frames. A receiving system calculates the location of the active area by interpolation based on the location of the active area for said some of the frames. In this way, the bandwidth requirements for communicating the location of the active areas is reduced.

Brief Description of the Drawings

Fig. 1 illustrates a display screen of a video editor/player used to annotate a video clip in accordance with the present invention.

Fig. 2 illustrates the position of an object on a video screen along the y direction vs. frame number for a sequence of frames in a video clip.

Fig. 3 is a flow diagram illustrating a method in accordance with a first embodiment of our invention.

Fig. 4 illustrates the data architecture for a video editor/player used to mark a video clip in accordance with the invention.

Fig. 5 is a block diagram of a signal transmission system and a reception and display system in accordance with another embodiment of the invention.

Fig. 6 illustrates the position of the active area along the y axis vs. the frame number of a video clip.

Detailed Description

Fig. 1 shows a display screen 100 depicting a typical graphical user interface ("GUI") for marking editors. Such an interface can be used during a method in accordance with our invention. Screen 100 shows a sequence of so-called key frames 101a to 101p in the editing area. As is known in the art, video images comprise a sequence of frames. For digital video, each frame corresponds to an array of pixels. Each pixel is a data value corresponding to the color, brightness and saturation of a small area of the frame. In certain video compression or encryption schemes, instead of storing every pixel array of every frame, the entire pixel array is stored only for certain "key" frames. Data is stored for the other frames that merely represent the difference between the image in those other frames and the preceding key frame. An example of a compression technique that uses this scheme is MPEG-2 (motion pictures expert group standard 2).

For the case of linearly digitized analog video that is not compressed or encrypted, every nth frame (e.g. every 10th frame) can be chosen as a key frame (along with frames corresponding to scene changes). As mentioned above, screen 100 depicts a sequence of key frames 101a to 101p.

Arrows 104a to 104d are not depicted on screen 100. Rather, arrows 104a to 104d are included in Fig. 1 merely to symbolize the chronological sequence of the frames.

Screen 100 includes a region 106 that depicts one of key frames 101a to 101p. The image in region 106 is magnified compared to the depiction of key frames 101a to 101p elsewhere on the screen, and represents the frame that is currently being edited. In the example of Fig. 1, key frame 101g is being edited. This is indicated by the presence of an icon or mark 101g' provided at frame 101g.

Typically, screen 100 also depicts navigation bars and toolbars for functions such as fast forward, fast backwards, new file, etc. The system driving display 100 includes software drivers for supporting the navigation bars and toolbars. These navigation bars and toolbars are not shown in Fig. 1 for sake of clarity.

Also shown on screen 100 is a menu box 108 which enables a user to select a shape of an active area about an object that a viewer will be able to click on. Box 108 enables an editor to select among many different shapes for the active area, only some of which are depicted in Fig. 1. As mentioned above, the active area is the location on a screen where an object of interest (e.g. a car or a person) is displayed. By clicking within the active area associated with that object or person, one effectively clicks on that object or person.

(When a viewer ultimately watches the video clip, the active area may or may not be displayed. For example, the system on which the video is displayed may highlight the active area. Alternatively, other systems on which the video is displayed may not highlight the active area. In yet other systems, the viewer may have the option to decide whether or not to highlight the active area.)

Referring back to Fig. 1, box 110 is a menu box that allows the editor to introduce the link into the image, which is typically a URL. However, other types of links can be provided, including specialized links for launching desired application software.

As seen in Fig. 1, region 106 depicts a person 112, whose head 113 has been marked by a square 114 and linked to a link L1. (The square shape has been selected in menu box 108, and link L1 has been selected in menu box 110.) Square 114 symbolizes the active area. During a scene in a video, objects (e.g. the head of a person) typically move in a limited, non-jerked manner. (Good camera control during filming of a movie, television show or other video typically involves minimizing any jerking of the object of interest on the screen. Also, a small object moving in a jerked fashion, even if closely followed, would be difficult for a viewer to target when viewing the annotated video.)

Square 114 represents the active area of the video screen on which a viewer will click if he wants to access information linked to person 112. For example, if the video is a soap opera, and the viewer wants to obtain more information about the character played by person 112, he will click on person 112, and in particular, he will position the cursor within the active area represented by square 114

During the video, person 112 typically moves about the set. Accordingly, it is necessary to have active area 114 move about the screen. This is necessary because if person 112 walks from the left side of the video screen to the right side of the video screen, and the viewer wants to find out more about person 112, he is going to want to click on current location of person 112.

In principal, one could annotate each frame of the video to determine where active area 114 should be positioned, but this would be an enormously tedious, time-consuming task. Accordingly, it would be desirable to simplify this task. In accordance with the present invention, some of the frames are annotated by placing active area 114 at an appropriate location. (This can be done manually or through other techniques.) The other frames are annotated by an interpolation method described below.

Fig. 2 illustrates the position of head 113 in a video along the y direction for each frame in a series of frames from a start frame A to an end frame D. Point 203 represents the starting position of head 113 at the start of a scene and point 204 represents the ending position of head 113 at the end of the scene. Line 206 traces the position of head 113 between points 203 and 204 as the scene progresses. (Head 113 changes position, e.g. as person 112 moves about the set during the video.)

Line 208 shows a direct linear interpolation, resulting by having a straight line between the two positions 203 and 204. In other words, rather than annotating each frame manually, the position of active area 114 is automatically inserted into each frame by performing linear interpolation between positions 203 and 204. As can be seen, this is not necessarily the most accurate way of annotating the frames. One can improve the linear interpolation by manually annotating more frames, and linearly interpolating between each frame for which a manual annotation has been performed.

Fig. 2 also shows intermediate points 216, 218, indicating the position of head 113 in frames B and C, respectively. In one embodiment, the position of active area 114 is manually inserted (e.g. visually) for a set of frames, e.g. frames A, B, C and D. The position of active area 114 is established by linear interpolation between each point for which the position of active area 114 has been manually annotated. Thus, in Fig. 2, lines 224, 226 and 228 are the result of linearly interpolating between points 203 and 216, points 216 and 218, and points 218 and 204 respectively.

A more accurate way of interpolating is to use an interpolation technique that results in the generation of a curve that represents the position of the object from frame to frame. An example of such a technique is spline interpolation. When performing spline interpolation, one typically only needs the end points and one or two intermediate points, and the spline interpolation technique will accurately track the object. The spline will smooth the motion and eliminate jitter often seen by image recognition.

It will be appreciated that one must not only establish the position of an object in the y direction, but also the x direction. Thus, during a method in accordance with our invention, the position of the active area will be established by interpolation for both the y and x directions. In addition, the object associated with the active area can expand or contract during the course of a video scene, e.g. because it moves closer to or further from the camera. Accordingly, in one embodiment, the size of active area 114 is also manually annotated for several frames during a scene. and the size of the active area is thereafter interpolated in the same manner as the position of the object.

Fig. 3 is a block diagram illustrating a method in accordance with a first embodiment of the invention. Referring to Fig. 3, the method begins by loading a block of video frames into a memory (step 301). The memory can be a magnetic memory (e.g. a magnetic disk, magnetic tape or magneto-optic disk). The memory can also be an optical disk or a semiconductor memory such as a DRAM, SRAM, EPROM, or other type of memory. The video information can be either in analog form or in digital form.

Thereafter, one manually annotates a subset of those frames (step 302). During this step of manual annotation, a set of frames are typically displayed on portions 101a to 101p of the screen 100, and one of those frames can be selected for annotation at a time. The selected frame is shown on portion 106 of screen 100. The location, size and shape of the active area are established within those frames that are being annotated, and a link associated with that active area is established. Manual annotation can be performed using an input device such as a mouse, touchpad, joystick, trackball, cursor control keys, or other input device of the video editing system. During this step, the annotation information is stored in a memory, e.g. as described below.

For the frames that are not manually annotated, annotation is done by interpolation (step 303). This can be done by any of the following steps:

a) linear interpolation of the position of the active area and linear interpolation of the size of the active area;

b) linear interpolation of the position of the active area with no change in size of the active area;

c) spline interpolation of the position of the active area and spline interpolation of the size of the active area; or

d) spline interpolation of the position of the active area with no change in the size of the active area.

In lieu of linear or spline interpolation, other interpolation techniques can be used. By way of example, interpolation can be:

a) between the start and end frames in a video clip; or

b) interpolation from each manually annotated frame to the next manually annotated frame. (Typically, the start and end frames of the clip are among the frames that are manually annotated.)

Referring to block 304, it is possible that interpolation can produce an error in position of the active region. In other words, there will be a difference between the position of an object and the position of the active area associated with that object. After interpolation, a person annotating the video clip can view the location of the active areas for the various frames at portions 101a to 101p of screen 100. If one of the frames shows a large discrepancy between the position of object 113 and active area 114, the person editing the video clip can manually annotate that frame, and have the editing system re-compute (e.g. by interpolation) all of the active area positions for the frames that have were not manually annotated.

After establishing the position of active area 114, the person annotating the video clip chooses the form of the links represented for users. Instead of using a box, circle, ellipse or other shape, the active area could by depicted as a logo (e.g. a GIF file), a piece of animation (e.g. an animated GIF file) or other video stream segment. (GIF is an acronym for graphics interchange format.) As mentioned above, optionally, the active area can be visually indicated when the annotated video clip is later viewed.

Referring to block 306, either after or before interpolation, a link to a URL can be associated with each active area. Thus, when a viewer clicks on that active area, that URL will be invoked. As mentioned above, the link need not be to a URL. It can also be a link to application software, or a source of information stored at a particular address.

Fig. 4 illustrates a data architecture for an embodiment of a video editor/player used to mark a video clip. Such a video editor/player comprises a data mass storage device 406 that contains numerous video clips. The data mass storage device can be a magnetic disk, a magneto-optic disk, an optical disk, a semiconductor memory, or other type of memory storage device.

Only one video clip 420a is shown in Fig. 4 for clarity, along with one matching annotation file 421a, although more than one video clip and annotation file may be stored in memory device 406. There can be more than one annotation file per video clip. The annotation file contains such things as the spline (either as a formula or as a reference to a predefined set of spline types) and its associated parameters, the link itself, and the color and size of the marker of the active area. Element 450 is a table or database, containing all the associated labels for each video sequence of frames, thus serving as a directory for the video presentation. There may be, of course, several databases of shows and annotation files, possibly organized in a hierarchy structure, and they may share some of the files 420 or 421 depending on the edits done.

In one embodiment, during annotation, data is loaded into a memory corresponding to the frame number, the location of the active area, its shape, its size, and the URL or other link associated with that active area. That data is provided to the viewer along with the video signal, typically in a separate file. This information can be provided as a separate data stream to the viewer. The viewer's video display system tracks the frame number, and for each frame, determines whether a cursor is positioned within one of the active areas established for that frame.

In another embodiment, information concerning the location of the active area, its shape, its size, and the link are communicated and embedded as part of the video signal. This information can be provided during one of the blanking intervals of the signal or during some other portion of the signal that is not typically used.

Embodiment In Which Interpolation Errors Are Minimized or Corrected

As can be seen in Fig. 2, when using interpolation to establish an active area on a video screen, there can be errors between the position of the marked object and the position of the active area. In accordance with another embodiment of the invention, these errors can be reduced or minimized through the following technique.

1. A block of video frames is loaded into a memory (step 301 of Fig. 3).

2. A first set of n frames is manually annotated, where n is greater than or equal to 2 (step 302 of Fig. 3). The n frames can be the first and last frames in a video sequence.

3. Interpolation is performed to establish an estimated position of an active area for those frames within the video sequence that are not manually annotated (step 303 of Fig. 3). Interpolation can be by linear interpolation. Alternatively, interpolation can be by a technique that results in a curve, e.g. a spline technique. As mentioned above, several interpolation techniques can be used to generate lines or curves, and the person annotating the video clip can choose among them.

4. For each frame that is not manually annotated, the video system calculates the distance between the active area and the object that the active area is

associated with. (The position of the active area is established by interpolation. The position of the object is established by automated image recognition techniques, e.g. as discussed above. As indicated above, image recognition may or may not be accurate.)

5. The system flags those frames that have the greatest distance between active area 114 and the object 113 that the active area is associated with. The system can highlight those frames (e.g. one or more frames 101a to 101p) that exhibit these distances and can place an icon or other marker at those frames, or it can use some other technique for identifying these frames to the user. The user can look at these frames, and verify whether the active area is, in fact. an unacceptable distance from the object with which it is associated.

6. If the user believes that this distance is unacceptable, he or she can manually annotate the frames exhibiting the unacceptable error (step 304 of Fig. 3).

7. After manually annotating the frames exhibiting the unacceptable error, the user can cause the system to recompute (e.g. by again interpolating) the position of the active area based on data including the new manual annotation.

8. Optionally, the user can have the system take the new interpolated active area positions and establish another list of frames having the largest errors. If the user finds these errors to be acceptable, or merely artifacts relating to image recognition errors, he or she can ignore them.

9. One then associates an active area shape (and optionally a visual indicator) and a link with the active area.

As mentioned above, during step 304 of Fig. 3, one can cause the system to compute the distance between the active area and the object associated with that active area, and flag those frames shown at portions 101a to 101p of screen 100 having unacceptable errors. The user can manually annotate those frames exhibiting unacceptable errors as described above. (This process can be used to adjust both the position (along both the x and y axes) and size of the active areas.)

Communicating Information Regarding the Active Areas to a Television Viewer

After the video frames are marked with information concerning the active areas, it is necessary to communicate that information to the viewer's video system. Fig. 5 illustrates a video transmission system 500 comprising a data mass storage device 506 including a portion 520 that stores a video program to be transmitted. The video program can be any type of program—sporting events, soap operas, educational programs, cartoons and other animation, etc. Also included within mass storage device 506 is a portion 521 that contains annotation information. This annotation information includes a) the location of several active areas associated with various frames of the video program; b) the size of those active areas; and c) the link that those active areas are associated with. (The link can be in the form of a URL.) In other words, the annotation information might indicate that in a particular frame of the video program, there is a first active area located in the upper left portion of the video frame, and that active area has a certain size. If the viewer clicks on that active area, that should activate a link to a particular web page.

Mass storage device 506 can be any type of mass storage device, e.g. a magnetic disk, magnetic tape, optical or magneto-optic media, or other data storage structure.

Information from mass storage device 506 is provided to a data separation circuit 505 that separates out the video signal and provides that video signal to a conventional transmitter 504. (Data separation circuit 505 is typically a server.) A television system 501 includes a receiver 511 for receiving the video signal. While Fig. 5 includes symbols indicating that the video signal is transmitted and received as conventional television radio signals, the video signals can be communicated by other means, e.g. an electrical cable, optical cable, microwave satellite communication, telephone lines, the internet, or other signal communication technique.

The annotation information (e.g. the location and size of the active area, along with the URL information associated with the active area) is communicated to televisions system 501 via a communication medium 503, which can be the same as, or different from, the medium by which the video signal is communicated. Thus, communication medium 503 can be conventional radio signals, an electrical cable, optical cable, microwave, satellite communication, telephone line, the internet, or other signal communication technique.

The annotation information is received by a preprocessing unit 512. Preprocessing unit 512 calculates, via an interpolation technique, the location of each active area of each frame for those frames in which that data was not provided by transmission system 500. In one embodiment, the interpolation technique is a linear interpolation technique, e.g. as described above with respect to which Fig. 2. For example, suppose transmission system 500 provides the location for an active area every tenth frame. Fig. 6 illustrates that the y coordinate values for the active area for frames 0, 10, 20 and 30, indicated by points 600, 601, 602 and 603, respectively. If a linear interpolation technique is used, it will be assumed that the y coordinate for the active area will be as established by lines 604, 605 and 606. Other interpolation techniques can be used, e.g. techniques that generate a curve instead of a straight line between specific points. For example, a spline interpolation technique can be used. If a spline interpolation technique is used, depending upon the spline calculation parameters, it will be assumed that the y coordinate for the active area will be along curve 607.

Fig. 6 illustrates the y coordinate of the active area vs. frame number. Similar calculations are performed for the x coordinate of the active area, and the size of the active area. (The size of the active area can increase or decrease, e.g. depending on whether the object associated with the active area moves toward or away from the camera.)

Referring again to Fig. 5, it is seen that the preprocessing unit 512 provides the video signal to a video display unit 514. Preprocessing unit 512 is coupled to a user interface. The user interface 513 permits a user to control certain features of the image displayed on display unit 514, e.g. whether a marker will be shown on the screen indicating the location of the active area. User interface 513 also permits a user to click on the various active areas to activate the links associated therewith.

User interface 513 typically has an appropriate control mechanism such as a trackball, mouse, touchpad or joystick. This interface is used to position a visible cursor on video display unit 514 so that the viewer knows what object he is clicking on. Preprocessing unit 512 determines whether the cursor has been positioned over an active area. In one embodiment, preprocessing unit 512 comprises a microprocessor system.

As mentioned above, in one embodiment of the invention. the active areas for selected frames are established, e.g. using a manual annotation system. The active areas are associated with objects depicted in the video frames. During this method, the location, size and shape of each active area is established for some (but not all) of the frames of the video clip. For other frames of the video clip, the location and size of the active areas is established by interpolating between the frames. In one embodiment, the location and size of the active areas for those frames that were manually annotated are communicated to video system 501. In other embodiments, the frames for which active locations and sizes are communicated are selected on other bases. For example, the location and size of the active areas can be communicated to receiving video system 501 for every tenth frame. Alternatively, the location and size of the active areas can be communicated only for the so-called "key frames" of the video clip. (As mentioned above, certain video compression algorithms communicate the entire pixel array only for certain key frames. For other frames, only information corresponding to the change between one frame and the next is communicated. The key frames are those frames for which the entire pixel array is communicated.)

Referring again to Fig. 5, in one embodiment, a workstation 550 can be coupled to video transmission system 500. Workstation 550 comprises a personal computer ("PC") 551 and a local memory storage device 552 (e.g. a disk) or other non-local storage (e.g. a server) that contains editing software 553. Workstation 550 is used to annotate the video clip. In this example, workstation 550 is coupled via a communication line 554 to server 505, but many other ways to network workstation 550 to server 505 can be used. In some embodiments, editor software 553 can run on server 505. In other embodiments, the files (both the video and annotation files) may reside on the local memory 552.

While the invention has been described with respect to specific embodiments, those skilled in the art will appreciate that changes can be made in form and detail without departing from the spirit and scope of the invention. For example, the pixels of the video frames can be stored in terms of RGB values. (RGB refers to the values of the red, blue and green color components associated with the pixel.) Alternatively, the pixel values can be stored in terms of color, saturation and intensity. Screen 100 and/or video display unit 514 can be a CRT screen, and LCD, a projection display, or other appropriate display technology. Different types of video editing/annotating systems can be used to practice the method of the present invention. The interpolation can be performed by a microprocessor or other computing device within the system.

The method of the present invention can be used to provide or receive more than one active area in a video clip.

Frames can be chosen for manual interpolation in different manners. For example, in one embodiment, every nth frame is manually interpolated, where n is an integer (e.g. 10).

The apparatus chosen to perform the interpolation and editing can be a microprocessor-based system with a display screen executing video editing and/or image recognition software programs.

In one embodiment, the first and last frames are not manually annotated. Accordingly, in such embodiments, the location and size of the active areas are extrapolated.

When the video signal is sent to a viewer, both the video and annotation information can be sent as separate signals to receiving system 501, or they can be communicated as parts of a combined signal. The combined signal can be provided by transmitter 504 via broadcast medium 502. Alternatively, the combined signal can be provided via any of the above-mentioned communications media, e.g. optical, cable, microwave transmission, satellite transmission, etc.

The various aspects of the invention can be practiced independently of one another. For example, the annotation apparatus of Fig. 1 can be used without using the signal communication apparatus of Fig. 5. Similarly, the communication apparatus of Fig. 5 can be used to communicate video clips annotated using methods other than the method described above with respect to Figs. 1 to 4. Accordingly, all such changes come within the present invention.

We claim:

1. A method for establishing the location of an active area within a video clip, said video clip comprising a set of frames, said method comprising the acts of:

(a) establishing the location of said active area for a first plurality of frames within said set of frames;

(b) establishing the location of said active area for a second plurality of said frames within said set of frames by interpolation based on the location established during act (a) above.

2. Method of claim 1 wherein the locations established during the acts (a) and (b) are stored in a memory device.

3. Method of claim 1 wherein act (a) comprises:

displaying said first plurality of frames on a display device; and

manually annotating said frames by storing within a memory information concerning the location of said active area within said first plurality of frames.

4. Method of claim 1 wherein said interpolation is linear interpolation.

5. Method of claim 1 wherein said interpolation is spline interpolation.

6. Method of claim 1 wherein said first plurality of frames are key frames.

7. Method of claim 1 wherein said interpolation is performed by a computing device within an editing system.

8. Method of claim 1 further comprising the acts of:

(c) displaying the active area for a third plurality of frames within said set of frames;

(d) annotating at least some of said frames within said second plurality; and

(e) establishing the position of said active area within said second plurality but not within said some of said frames by interpolation after performing act (d).

9. Apparatus comprising:

a memory device storing a video clip, said video clip comprising a plurality of video frames;

an input device for inputting information into said apparatus indicating the location of an active area within some of said video frames; and

a computing device within said apparatus for establishing the location of said active area for other frames within said plurality.

10. Apparatus of claim 10 wherein said computing device establishes the location of said active area for said other frames within said plurality of interpolation.

11. A method comprising:

providing a video clip comprising a set of frames, a first plurality of frames within said set being annotated with information indicating the location of an active area, said active area being associated with an object shown within said video clip;

establishing location data for said active area for those frames that are not within said first plurality by interpolation;

using automated image recognition to identify one or more of said frames that are not within said first plurality for which the distance between said active area and said object is large compared to others of said frames that are not within said first plurality.

12. Method of claim 11 wherein said automated image recognition identifies frames in which the distance between said active area and said object exceeds a threshold value.

13. Method of claim 11 wherein said automated image recognition identifies the n frames for which the distance between the active area and object are largest, where n is an integer.

14. Method of claim 11 further comprising displaying said one or more frames on a display so that a viewer can see the distance between said active area said object in said one or more frames.

15. Method of claim 11 further comprising annotating at least one of said one or more frames after said act of using.

16. Method of claim 15 further comprising establishing location data for said active area for those frames that are not within said first plurality and are not within said at least one frame by interpolation.

17. Method of claim 16 further comprising displaying the frames whose location data has been established by interpolation so that a user can see how far the active area is from the object.

18. Method of claim 11 wherein an operator annotates said first plurality of frames with said information prior to said act of establishing.

19. Method of claim 18 wherein said annotation is performed by storing data in a memory indicating the position of said active area for said first plurality of frames.

20. Method of claim 11 wherein said video clip is stored in a memory device.

21 Method of claim 20 wherein said information indicating the location of an active area is stored in the same memory device as said video clip.

22. Method of claim 21 wherein said video clip and said information indicating the location of an active area are located in different files within said memory device.

23. Method of claim 21 wherein said information indicating the location of said active area is embedded within said video clip.

24. Method comprising the steps of:

providing a video clip, said video clip comprising a group of frames, said video clip depicting an object;

annotating some of said frames within said group of frames, but not others of said frames within said group of frames, by storing in a memory the location of an active area within said some of said frames associated with said object;

establishing the location of said active area for said others of said frames by interpolation;

using automated image recognition to determine those frames within said others of said frames for which the distance between said active area and said object is large compared to others of said frames that are not within said first plurality.

25. Apparatus comprising:

a memory structure for storing a video clip, said video clip comprising a set of frames, a first plurality of frames within said set being annotated with information indicating the location of an active area, said active area being associated with an object shown within said video clip;

means for establishing location data for said active area for those frames that are not in said first plurality by interpolation;

means for using automated image recognition to identify one or more of said frames for which the distance between said active area and said object is large compared to others of said frames that are not within said first plurality.

26. A method comprising the steps of:

receiving a video image;

displaying said video image as a sequence of frames;

receiving information corresponding to the location of at least one active area for some of said frames but not others of said frames; and

determining the location of said active area for said others of said frames based on said received information.

27. Method of claim 26 wherein said step of determining is performed by linear interpolation.

28. Method of claim 26 wherein said step of determining is performed by spline interpolation.

29. Method of claim 26 wherein information corresponding to the size of said active area is received for said some of said frames but not said others of said frames, said method also comprising the step of determining the size of said active area for said others of said frames by interpolation.

30. Method of claim 26 wherein said active area is associated with a link, said method further comprising the step of activating said link.

31. Method of claim 26 wherein said link is a link to an internet page.

32. Method of claim 26 further comprising activating said link.

33. Method of claim 32 wherein said displaying comprises displaying said video on a display screen and said activating comprises clicking on an active area on said display screen.

34. Method comprising:

communicating a video clip comprising a set of video frames, some of the video frames within said set of video frames being annotated such that an active area is established for said some of said video frames;

communicating the location of said active area for said some of the video frames within said set of video frames and not others of said video frames within said set of video frames.

35. Method of claim 34 wherein said video clip is communicated through a first communication medium and said location of said active area is communicated through a second communication medium that is different from said first communication medium.

36. Method of claim 34 further comprising:

receiving said video clip and the location of said active area for some of said video frames within said set of video frames and not said others of said video frames; and

establishing positions for said active area within said others of said video frames by interpolation.

37. Method of claim 36 wherein said active area is associated with an object, said object being shown in each of said video frames within said set of video frames.

38. A system comprising:

a receiver for receiving a video signal;

a display coupled to said receiver for displaying a video image corresponding to said video signal, said video image comprising a set of frames;

a circuit receiving information indicating the position of at least one active area for some of said frames and not others of said frames, said circuit determining the location of said activc area for said others of said frames by interpolation based on the location of said active area for said some of said frames.

39. System of claim 38 further comprising a transmission system for transmitting said video signal and said information corresponding to the position of at least one active area for said one of said frames and not said others of said frames.

40. Apparatus comprising:

a first memory storage device storing a video signal, said video signal comprising a set of frames;

a memory storage device storing the location of an active area for some of said frames;

a transmitter for transmitting said video signal and said location of said active area for said some of said frames but not others of said frames, whereby a system receiving said video signal and said location of said active area for said some of said frames can establish a location for said active area for said others of said frames by interpolation.

41. Apparatus comprising:

a memory storing a video clip;

a structure for establishing the location of an active area for some of the frames of the video clip but not other frames of the video clip; and

a transmission system for transmitting the video clip and the location of the active area for said some of said frames of the video clip but not other frames of the video clip.

42. Apparatus of claim 41 further comprising a receiving unit for receiving and displaying the video clip and establishing the location of said active area for said other frames by interpolation.

43. A method for establishing the location of an active area within a series of frames of images, said method comprising the acts of:

(a) establishing the location and/or size of said active area for a first plurality of frames within said series of frames;

(b) establishing the location and/or size of said active area for a second plurality of said frames within said series of frames by calculation based on the location established during act (a) above.

44. Apparatus of claim 40 wherein said memory storage device stores the location of said active area for some of said frames but not others of said frames.

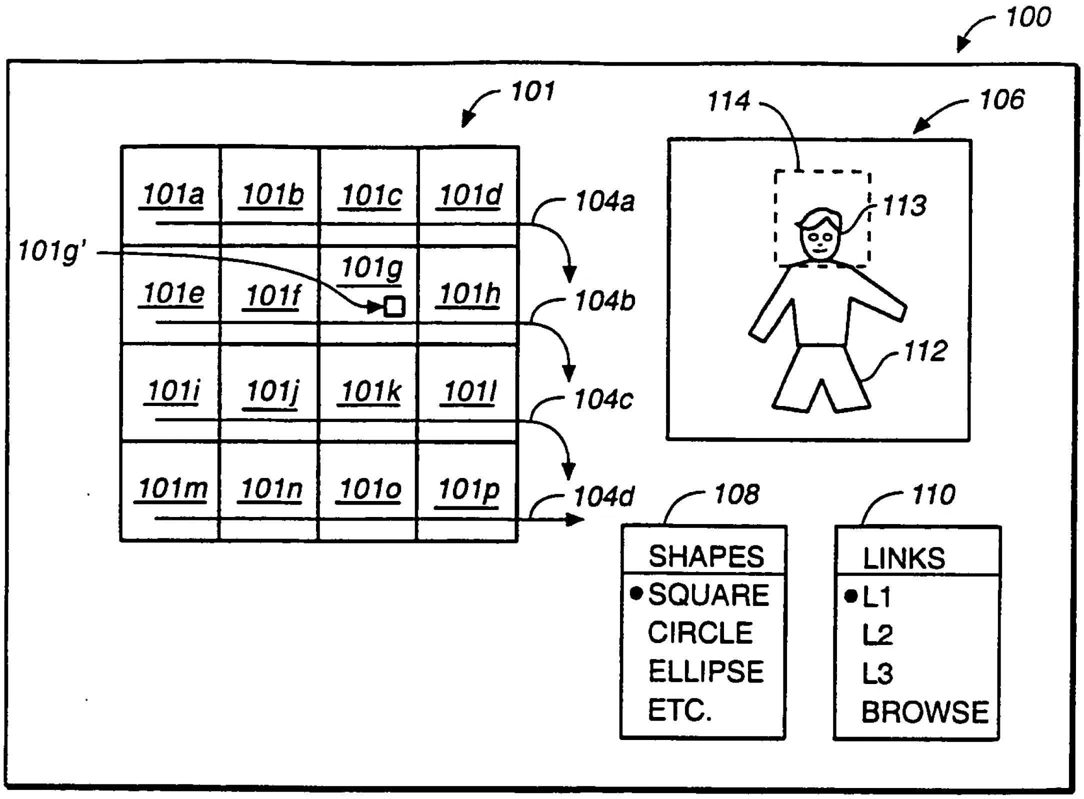

100
101
106
114
113
112
101a
101b
101c
101d
101e
101f
101g
101h
101i
101j
101k
101l
101m
101n
101o
101p
101g'
104a
104b
104c
104d
108
110
SHAPES
● SQUARE
CIRCLE
ELLIPSE
ETC.
LINKS
● L1
L2
L3
BROWSE

FIG._1

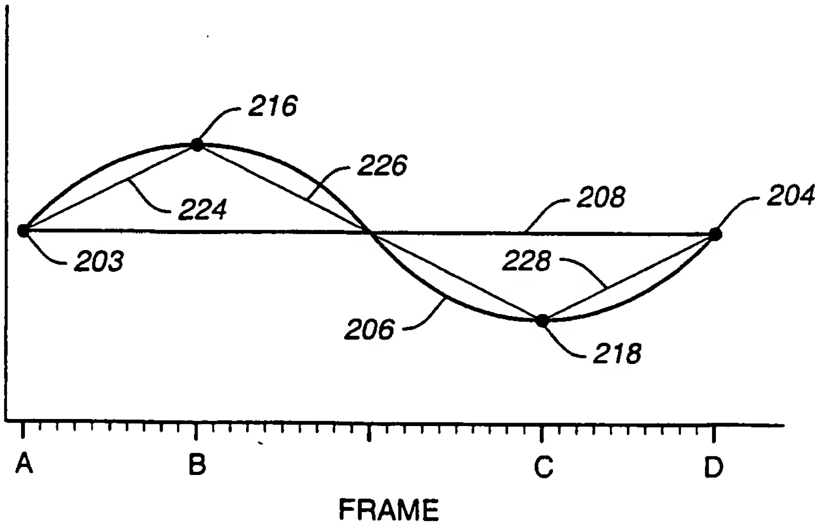

216
226
224
208
204
203
228
206
218
A
B
C
D
FRAME

FIG._2

FIG._3

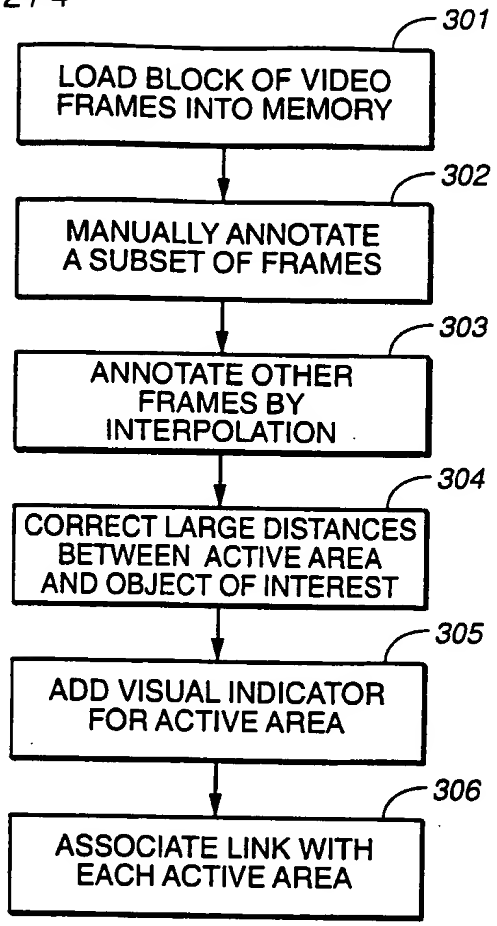
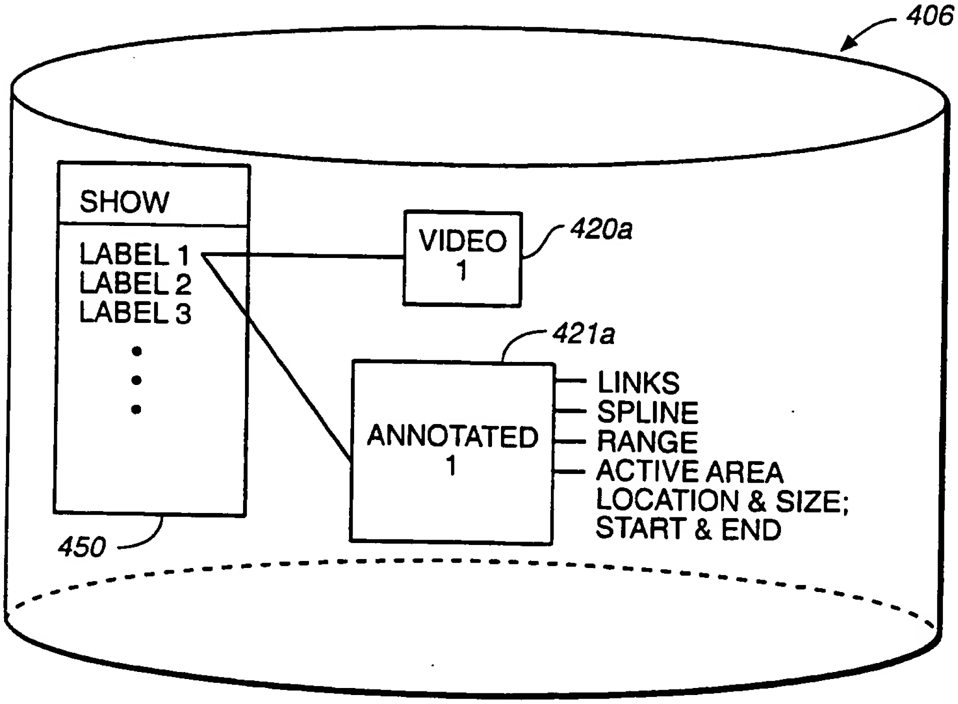

301
LOAD BLOCK OF VIDEO FRAMES INTO MEMORY
302
MANUALLY ANNOTATE A SUBSET OF FRAMES
303
ANNOTATE OTHER FRAMES BY INTERPOLATION
304
CORRECT LARGE DISTANCES BETWEEN ACTIVE AREA AND OBJECT OF INTEREST
305
ADD VISUAL INDICATOR FOR ACTIVE AREA
306
ASSOCIATE LINK WITH EACH ACTIVE AREA

FIG._4


406
SHOW
LABEL 1
LABEL 2
LABEL 3
VIDEO 1
420a
421a
ANNOTATED 1
LINKS
SPLINE
RANGE
ACTIVE AREA
LOCATION & SIZE;
START & END
450

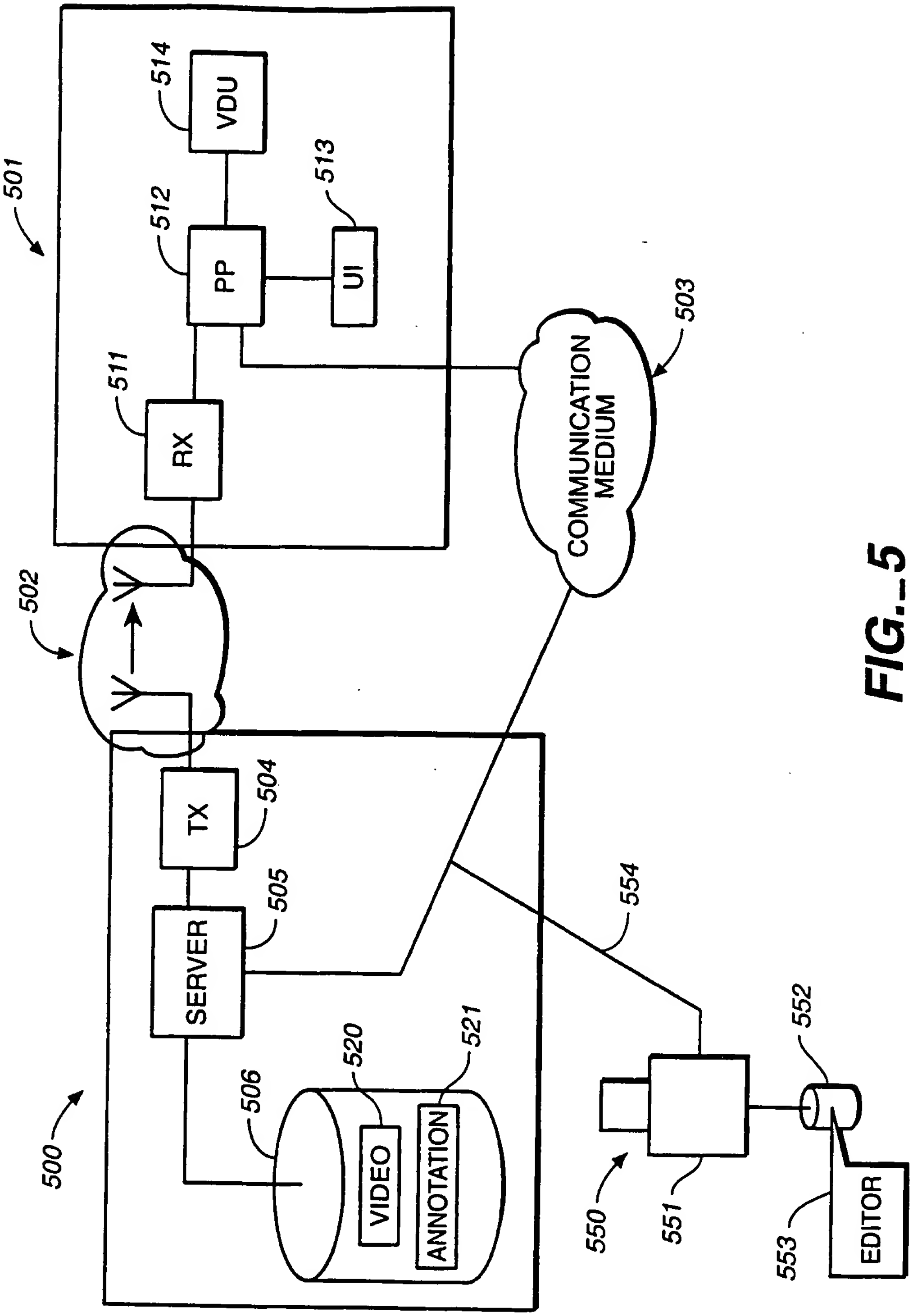

501
VDU
514
PP
512
UI
513
RX
511
COMMUNICATION MEDIUM
503
502
TX
504
SERVER
505
554
500
506
VIDEO
520
ANNOTATION
521
550
551
552
553
EDITOR

FIG._5

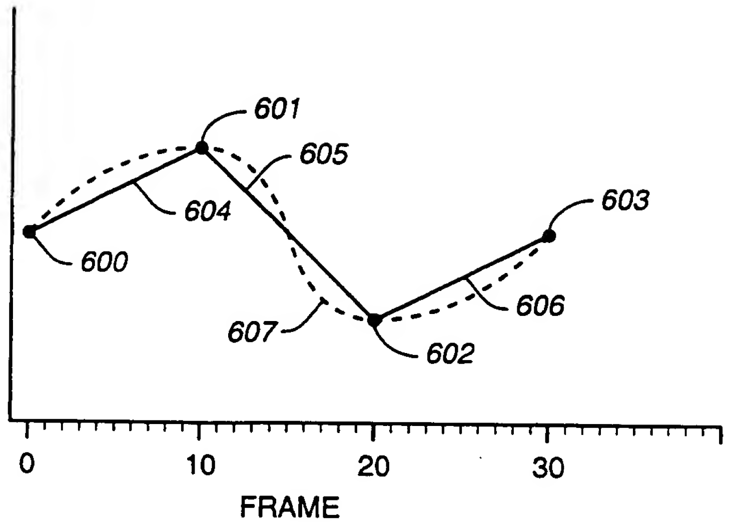


FIG._6

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/24263

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) :G06F 17/00
US CL :345/473, 474

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 345/473, 474, 418, 427

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X,P	US 6,006,241 A (PURNAVEJA et al) 21 December 1999, col.4, line 9 through col.9.	1-44
A	US 5,675,752 A (SCOTT et al) 07 October 1997.	1-44
A,P	US 6,032,156 A (MARCUS) 29 February 2000.	1-44

□ Further documents are listed in the continuation of Box C. □ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search 02 DECEMBER 2000	Date of mailing of the international search report 09 JAN 2001
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer CLIFF NGUYEN VO Telephone No. (703) 305-9594

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
21 December 2000 (21.12.2000)

PCT

(10) International Publication Number
WO 00/77664 A2

(51) International Patent Classification⁷: G06F 17/00

(21) International Application Number: PCT/US00/15938

(22) International Filing Date: 8 June 2000 (08.06.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/330,904 11 June 1999 (11.06.1999) US

(63) Related by continuation (CON) or continuation-in-part (CIP) to earlier application:
US 09/330,904 (CON)
Filed on 11 June 1999 (11.06.1999)

(71) Applicant *(for all designated States except US)*: LIBERATE TECHNOLOGIES LLC [US/US]; 2 Circle Star Way, San Carlos, CA 94070-6200 (US).

(71) Applicant *(for US only)*: BOUCHER, Antoine [CA/CA]; 32-270 North Centre Road, London, Ontario N6G 5E2 (CA).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: MCRAE, Paul, E. [CA/CA]; 141 Chesham Place, London, Ontario N6G 3T7 (CA). SCHEYEN, Peter, G., N. [CA/CA]; 20 Inverary Road, London, Ontario N6G 3L6 (CA).

(74) Agent: BRUCKNER, John, J.; Wilson Sonsini Goodrich & Rosati, 650 Page Mill Road, Palo Alto, CA 94304-1050 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *Without international search report and to be republished upon receipt of that report.*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: METHODS, APPARATUS, AND SYSTEMS FOR STORING, RETRIEVING AND PLAYING MULTIMEDIA DATA

(57) Abstract: Various embodiments of the invention provide increased speed and decreased computer processing for playing and navigating multimedia content by using two types of data objects for displaying the multimedia content. The data object type includes rendered multimedia content data. The second data object type provides semantic content corresponding to the rendered multimedia content. The storage medium in which these two types of data objects are contained is referred to as a rendered cache. The semantic content can include locations, sizes, shapes, and target universal resource identifiers of hyperlinks, multimedia element timing, and other content play instructions. The very fast play of content stored in the rendered cache is due to the elimination of the steps of laying out the content, rendering the content, and generating the semantic representation of the content. These steps are required each time the content is played after retrieval from a conventional cache. The only steps required for playing content from the rendered cache are to read the rendered content, read the semantic content, restore the semantic representation, and play the content. A traditional web browser visiting a web site that resides in a rendered cache provides an almost instantaneous display of the web site. The caching mechanism provided by various embodiments of the invention is independent of content file format and the stored semantic content file format. As long as a client application, such as a content browser, can recognize and play the multimedia content and recognize and interpret the semantic content, the application can realize the benefits provided by the rendered cache.

METHODS, APPARATUS, AND SYSTEMS FOR STORING, RETRIEVING AND PLAYING MULTIMEDIA DATA

BACKGROUND OF THE INVENTION

Field of the Invention

The invention relates generally to improvements in computer systems. More particularly, the invention relates to methods, apparatus, and systems storing multimedia content such as audio, text, image, and graphical content in a cache directory.

Discussion of the Related Art

Prior art graphics processing storage medium, sometimes called a cache system, is known to those skilled in the art. For example, a conventional caching system is typically composed of a small fast storage device that contains a "snapshot" of information originally received from a larger, slower source. The snapshot is considered by the particular implementation to be the most relevant information to the processing occurring during the current time period.

In the context of Internet content, a "cache" is a file, database, directory, or set of directories disposed in a computer file system. The cache stores content that has been previously retrieved generated or otherwise produced. Internet browsers and editors use cache directories to store content. The cached content is used in place of remote content whenever possible in order to decrease retrieval latencies. Therefore, many web browsers and text editors save Internet and other text and graphical content in a cache directory in order to reduce access times. This content is usually stored in its original form [for example, hypertext markup language (HTML) and accompanying images].

A problem with this technology has been that to view content based on the image data stored in the cache typically requires layout and rendering of the data. If the data upon which content is based does not change, the process of rendering need only occur once to a display buffer. When information is changed, the information must be re-rendered to reflect the desired change. For complex graphics scenes re-rendering can require massive processing for only incremental changes in the scene or particular graphic. The layout and rendering processes are time consuming and require processor resources. Therefore, what is required is solution that provides play of multimedia content more efficiently in terms of time and processor resources.

Heretofore, the requirements of timely and processor efficient play of multimedia content have not been fully met. What is needed is a solution that simultaneously addresses these requirements. The invention is directed to meeting these requirements, among others.

SUMMARY OF THE INVENTION

A primary goal of the invention is to provide timely and processor efficient display of multimedia content. In accordance with these goals, there is a particular need for a storage medium that includes multimedia content and the semantic content of the multimedia content. A storage medium including both the multimedia content and the semantic content is referred to herein as a rendered cache.

For various embodiments of the invention, the semantic content can include locations, sizes, shapes, and target universal resource identifiers of hyperlinks, multimedia element timing, and other content play instructions. The very fast play of content stored in the rendered cache is due to the elimination of the steps of laying out the content, rendering the content, and generating the semantic representation of the content. These steps are required each time the content is played after retrieval from a conventional cache. The only steps required for playing content from the rendered cache are to read the rendered content, read the semantic content, restore the semantic representation, and play the content.

A traditional web browser visiting a web site that resides in a rendered cache provides an almost instantaneous display of the web site. The caching mechanism provided by various embodiments of the invention is independent of content file format and the stored semantic content file format. As long as a client application, such as a content browser, can recognize and play the multimedia content and recognize and interpret the semantic content, the application can realize the benefits provided by the rendered cache. Thus, it is possible to simultaneously satisfy the above-discussed requirements of timely and processor efficient display of multimedia content, which, in the case of the prior art, are not simultaneously satisfied.

A first aspect of the invention is provided as an embodiment that is based on a method, implemented in at least one computer, for storing multimedia data. The method for storing multimedia data comprises detecting multimedia content, generating a semantic representation of a rendered representation of the multimedia content from the play instructions, storing the rendered representation in a storage medium, and storing data corresponding to the semantic representation in the storage medium. The multimedia content includes play instructions and at least one multimedia element. The at least one multimedia element includes at least one of graphical images, audio, text, and full motion video. The play instructions include at least one of timing of the multimedia content and ordering of the multimedia content. The semantic representation describes at least one of characteristics of the rendered representation, and relationships between different multimedia elements disposed in the rendered representation.

A second aspect of the invention is provided as an embodiment that is based on a method, implemented in at least one computer, for storing multimedia data. The method for storing multimedia data comprises detecting multimedia content including layout instructions, and laying out the multimedia content according to the layout instructions to form rendering instructions and a semantic representation of a rendered representation of the multimedia content. The method also includes rendering the multimedia content according to the rendering instructions to produce the rendered representation, storing the rendered representation in a storage medium, and storing data corresponding to the semantic representation in the storage medium.

A third aspect of the invention is provided as an embodiment that is based on a method, implemented in at least one computer, for retrieving multimedia data. The method for retrieving multimedia data comprises processing resources of a first computer of the at least one computer detecting a request for requested multimedia content, and processing resources coupled with the first computer determining whether data corresponding to the requested multimedia content is disposed in a storage medium. The storage medium is coupled with the first computer and includes rendered representations of multimedia content and semantic content. Embodiments according to the third aspect of the invention also include responding to a determination that data corresponding to the requested multimedia content are disposed in the storage medium by retrieving a rendered representation of the requested multimedia content; and retrieving semantic content corresponding to the requested multimedia content.

A fourth aspect of the invention is implemented in an embodiment that is based on a rendered cache comprising a storage medium, and an indexing mechanism adapted to store and retrieve a rendered representation of the multimedia content formatted for rapid play and semantic content of the multimedia content.

A fifth aspect of the invention is implemented in an embodiment that is based on a client. The client comprises processing resources adapted to detect a rendered representation of multimedia content and semantic content of the rendered representations, and processing resources adapted to respond to detecting the rendered representation of the multimedia content and the semantic content by playing at least a portion of the rendered representation according to the semantic content.

A sixth aspect of the invention is implemented in an embodiment that is based on a system for using multimedia content. The system comprises web crawler processing resources adapted to access the multimedia content from source data storage, rendering processing resources, and a rendered cache as described above as the fourth aspect of the invention. The rendering processing resources are adapted to generate a semantic representation of a rendered representation of the multimedia content, and format the semantic representation as semantic content, and render the multimedia content into the rendered representation, the rendered representation is formatted for rapid play.

A seventh aspect of the invention is implemented in an embodiment that is based on a system for accessing multimedia content. The system for accessing multimedia comprises a rendered cache as described above as the fourth aspect of the invention, and rendering processing resources adapted to convert the multimedia content into the rendered representation, the rendered representation is formatted for rapid play, and create a graphical representation of the multimedia content.

An eighth aspect of the invention is implemented in a method for playing multimedia content. The method comprises retrieving a rendered representation of the multimedia content from a storage medium, and retrieving semantic content of the rendered representation from the storage medium. The method includes browser processing resources reading the rendered representation and the semantic content, and the browser processing resources restoring a semantic representation based on the semantic content. The method includes the browser processing resources transmitting an active portion of the rendered representation to a client, and transmitting an active portion of the semantic content corresponding to the active portion of the rendered representation to the client. The active portion of the rendered representation is one of a portion of the rendered representation presently being played, and a portion of the rendered representation to be played rapidly after transmitting. The method also includes client processing resources detecting the active portion of the rendered representation and the active portion of the semantic content, and the client processing resources playing the active portion of the rendered representation.

BRIEF DESCRIPTION OF THE DRAWINGS

These, and other, goals and aspects of the invention will be better appreciated and understood when considered in conjunction with the following description and the accompanying drawings. Various embodiments of the invention are illustrated in the drawings accompanying and forming a part of this specification, wherein like reference characters (if they occur in more than one view) designate the same parts. It should be noted that the features illustrated in the drawings are not necessarily drawn to scale.

FIG. 1 illustrates a schematic block diagram of a conventional method for retrieving and playing multimedia content, appropriately labeled "PRIOR ART".

FIG. 2 illustrates a schematic block diagram of a process overview for retrieving and playing multimedia content using a rendered cache, representing an embodiment of the invention.

FIG. 3 illustrates a schematic block diagram including render process details, representing an embodiment of the invention.

FIG. 4 illustrates a schematic block diagram including play process details, representing an embodiment of the invention.

FIGS. 5A-5B illustrate screen shots of portions of a Toronto Exchange Internet page, representing an embodiment of the invention.

FIG. 6A illustrates the timing of play of different multimedia elements for an example of multimedia content that does not require layout, representing an embodiment of the invention.

FIGS. 6B-6D illustrate different images included in the multimedia content not including layout example representing an embodiment of the invention.

FIG. 7 illustrates a communications system including a rendered cache, representing an embodiment of the invention.

DESCRIPTION OF PREFERRED EMBODIMENTS

The invention and the various features and advantageous details thereof are explained more fully with reference to the non-limiting embodiments that are illustrated in the accompanying drawings and detailed in the following description of preferred embodiments. Descriptions of well-known components and processing techniques are omitted so as not to unnecessarily obscure the invention in detail. It should be understood, however, that the following description, while indicating preferred embodiments of the invention and numerous specific details thereof, is given by way of illustration and not of limitation. Many changes and modifications may be made within the scope of the invention without departing from the spirit thereof, and the invention includes all such modifications.

Methods, apparatus and systems are described for storing multimedia content using a process for caching fully rendered documents in a way that significantly increases content viewing speeds, navigation in a hyperlink document, while decreasing processing requirements.

Definitions

The following terms are used in the description of various embodiments of the invention provided herein.

Content: Text and graphical information that require a layout and/or rendering process in order to be viewed on a computer, television or other display device. Other terms for content include web-page, document, Internet content, hypertext markup language (HTML), eXtensible Markup Language (XML), and Television Markup Language (TVML). Content can also include non-graphical information such as audio.

Content Browser: A computer program designed to retrieve, display or navigate content. Examples include Internet web browsers, HTML/XML/Standard Generalized Markup Language (SGML) editors, word processors, and Internet web proxies.

HTML: The de facto Internet content standard. HTML includes a set of markup rules that describe the layout of Internet content. Browsers use this markup to layout and render the HTML for viewing computer monitors, televisions, or other displays.

Markup: Notation used to describe the syntactic and semantic features of a content document.

Multimedia Content: Multimedia elements used for playing a presentation for a user. The multimedia elements can include graphical images (including rendered HTML), audio, text, and full motion video.

Navigation: The process of selecting an indexing indication, such as a URI in the form of a hyperlink, from displayed content to access further content.

Presentation: Content that references at least one multimedia element. Presentations include play instructions that can be used to define the timing, order, and position of the multimedia plays. The play instructions can include the size, shape and target of all hyperlinks, information on interactive elements (like HTML forms), and Meta values.

Render: The process of generating a graphical representation of data that can be viewed on a display. For example, web browsers render HTML pages into graphical images that can be viewed on a computer monitor or television. Also the process of generating or converting multimedia data (images, audio, text, full motion video) into a format that can be played.

Rendered Cache: Various embodiments of the invention use the concept of a rendered cache to mean a cache of content that is not only generated (or retrieved) from a multimedia content data source, such as the Internet; but also is rendered and ready for rapid play. The rendered cache can include two types of objects: multimedia content and semantic content. The multimedia content stored in the rendered cache is content that has been rendered and is ready for very quick display. Semantic content includes a description of the semantic features or representation of the rendered content. Examples of semantic features include the location, size, shape and target of hyperlinks, the timing, location, and size of animated graphics interchange format (GIF) frames, the size and relative location of HTML frames, information on HTML forms, HTML meta values, presentation play timing, and other play instructions. A more detailed description of the rendered cache is provided in the Process Description section below.

Semantic Representation: A description of the characteristics, attributes, logical structure, and features of multimedia elements (or objects) that form a rendered representation of multimedia content, or a portion thereof. The data can also describe the relationships between different multimedia elements within a particular presentation portion, and the way various elements of the multimedia content are accessed and manipulated. The semantic representation is typically generated during the layout process and is structured such that the semantic representation can be saved as formatted and indexed semantic content in a file or database, and rapidly restored from the semantic content. The semantic content can be stored along with the multimedia content or as one or more separate indexed files. The semantic representation is independent of the format of the stored semantic content. The Document Object Model (DOM) is one type of semantic representation and is adapted for use with HTML and XML documents.

TVML: Some embodiments of the invention (including the VirtualModem™ presentation system provided by Interactive Channel, Inc. located in London, Ontario, Canada) use an XML language called television markup language (TVML) to describe multimedia content. TVML includes markup to describe how to play multimedia content. The multimedia content can include text (including HTML), graphical images, audio, text, and full-motion video. TVML can include markup to describe when each multimedia component should be played relative to the other multimedia components.

URI: A Universal Resource Identifier (or URI) is an Internet standard term for all types of names and addresses that refer to content. The term URI encompasses terms such as filename, hyperlink, and Universal Resource Locator (URL).

VMML: An XML markup language (called VMML - VirtualModem™ Markup Language) used to store semantic representations of rendered multimedia content by various embodiments of the invention, such as the VirtualModem™ presentation system.

XML: A markup language used to describe other markup languages, such as HTML and TVML.

Process Description

Various embodiments of the invention include methods, implemented in at least one computer, for storing and retrieving multimedia data. These methods navigate and play multimedia content with increased speed and decreased computer processing by using different types of data objects to represent the multimedia data. A first data object type includes pre-rendered multimedia content data. A second data object type includes a semantic representation of the pre-rendered multimedia content. These data object types can be stored as separate files or can be contained in the same file.

Prior art methods for retrieving and playing multimedia content are represented by Figure 1, which includes a traditional cache 110. After detecting a request to play multimedia content (at step 120), retrieving processing resources, such as those disposed in a web browser, retrieve the corresponding multimedia content data. A traditional web browser, such as Netscape Navigator, Netscape Communicator, or Microsoft® Internet Explorer, when coupled with a traditional cache 110, then performs the steps described below in response to each and every play request 120.

After retrieving the content (e.g., the HTML content description), the content is read (step 130) from either a traditional cache 110, an Internet 105, or another content data source. Processing resources disposed in a computer can layout the content (step 140), e.g., according to the content's HTML description. During the layout 140, the processing resources generate rendering instructions 140A and derive a semantic representation 140B of the multimedia content. Note that for some embodiments, layout 140 is not required. For these embodiments, the semantic representation 140B can be generated from play instructions, as shown in Figure 3 (at step 315).

Content browsers can use the semantic representation 140B to determine location, size, shape and targets of hyperlinks; and content play instructions. The semantic representation 140B can also be used to describe other interactive presentation elements, e.g., HTML forms. The semantic features corresponding to the depicted graphical representation generated for play when using traditional content browsers coupled with traditional caches 110 persist only as long as the content is being viewed. Because the semantic features must be present whenever the multimedia content is played, and because traditional caches 110 store the multimedia content in a non-rendered original form, traditional browsers must re-render the graphical representations each time a user requests the content, as shown in Figure 1.

A render engine then renders the multimedia content (at step 150) according to the rendering instructions 140A to form rendered content 160 (otherwise referred to herein as the rendered representation of the multimedia content). Finally, a multimedia play engine uses both the rendered content 160 and the semantic representation 140B to play the rendered content (at step 170). For multimedia content including images, the playing 170 includes displaying the rendered image on a user screen according to the semantic representation 140B. The rendered content 160 is also referred to herein as a rendered representation of the multimedia content.

In prior art implementations, playing 170 occurs after layout 140 (or other process in which the semantic representation 140B is generated), and rendering 150 have been completed. Rendering 150 and generation of the semantic representation 140B require a relatively long time between when the play request 120 is received and when multimedia content is played 170 compared to the time required to play content using various method embodiments of the invention. Also, greater data processing is required for rendering 150 and generation of the semantic representation 140B for the multimedia content than the processing required for play 170 using various method embodiments of the invention.

The details of retrieving (step 260), rendering 150, and playing 170 multimedia content for some embodiments of the invention are illustrated in Figs 2 through 4. Fig. 2 provides an overview of the retrieving 260, and the playing 170 processes for multimedia content, e.g., HTML content, using a rendered cache 201. The methods can be implemented in at least one computer having one or more programs for retrieving and playing multimedia content. The benefits of using the rendered cache 201 for subsequent access to the same multimedia content are also described below.

The rendered cache 201 includes not only rendered content 160 (which can include image data) but also some means of reconstructing the semantic representation 140B of the multimedia data. The reconstruction of the semantic representation 140B can be done using proprietary image formats or separate files that describe the semantic features. This semantic representation 140B can include locations, sizes, and destinations of hyperlinks, descriptions of animations or other dynamic content, and other "meta" information. Meta information can include tagging, refresh (client pull replacement), Meta lists, and platform for Internet content selection (PICS) association labels.

Some embodiments of the invention (including VirtualModem™ interactive presentation systems provided by Interactive Channel Technologies, Inc. located in London, Ontario, Canada) use an XML language called VMML to store the semantic content. The VMML semantic content can include markup to represent the following semantic features of the rendered content 160:

1. Location, size, shape, and target indices (such as URI) of hyperlinks,
2. Size and relative location of HTML frames in the rendered image,
3. Size, location, and timing of animated GIFs,
4. Size, location, and type of HTML form elements,
5. Timing of multimedia content elements, and
6. Other play 170 instructions.

Proper use of content from a rendered cache 201 eliminates the steps of generating a semantic representation 140B, layout 140 (when needed), and rendering 150. On the other hand traditional web browsers using traditional caching mechanisms must perform these steps before playing 170 the content. Eliminating these steps reduces the time and use of processing resources required for playing 170 the multimedia content.

A rendered cache 201 can include of two types of data objects: multimedia content and semantic content. The content can be stored 320 in any format (i.e., the caching mechanism is independent of file format). Typically, the layout 140 and/or rendering 150 processing resources format the semantic presentation 140B for storage in the rendered cache 201 as semantic content. Alternatively, the layout 140 and/or rendering 150 processing resources can transfer the semantic presentation 140B to rendered cache 201 server processing resources which then format the semantic presentation into semantic content to be stored in properly indexed files for retrieval 260. For some embodiments of the invention, content browsers (and/or other client applications using content from the rendered cache 201) can include processing resources, such as a program, for detecting the format of the rendered content 160 and for viewing multimedia content.

When a request for content is received (step 210) the content browser can determine (step 220) whether a rendered representation of the content already exists in the rendered cache 201. The browser can also determine (step 230) whether the content in the rendered cache 201 is outdated. The content request received at step 210 can be provided to the browser using a file target index, such as a Universal Resource Indicator (URI).

Once it has been determined whether an updated rendered representation of the requested data already exists in the rendered cache 201, a first and simpler processing path indicated in Figure 2 can be followed. Because the rendered cache 201 contains valid rendered content corresponding to the request, browser engine processing resources can simply read the semantic content and the rendered content 160, restore the semantic representation (step 240), and then play 170 the rendered content corresponding to the requested content.

The process proceeds along a second path if the server-based system (or other processing resources coupled with the rendered cache 201) has determined that the requested content is not in the rendered cache 201, or that the content stored in the rendered cache is outdated. Along the second path, the browser submits a request to retrieve the content from an updated source (e.g., the Internet 105) and retrieves the content (step 260). After the updated content has been stored 320 in the rendered cache 201 (as shown in Figure 3), the process continues along the first method path as long as the stored content does not become out of date. The first method path, as shown in Figure 2, includes reading the semantic content and the rendered content 160, and restoring the semantic representation (step 240), to play 170 the rendered content for each request -

Figure 3 illustrates a more detailed depiction of the layout 140 and storing 320 processes. After retrieving 260 the requested multimedia content with layout instructions and/or play instructions, the computer determines whether layout 140 is required for the multimedia content (step 310). The semantic representation 140B of the semantic features is generated during the layout 140 process, or generated from play instructions (step 315) when no layout is required.

After rendering 150, the rendered content 160 is stored 320 in the rendered cache 201. Similarly, after construction of the semantic representation 140B, the semantic representation is formatted as semantic content and also stored 320 in the rendered cache 201.

If the rendered cache 201 stored only the resulting rendered content 160, the description of the hyperlinks, display instructions and other semantic content would be lost. The semantic content can take the form of flat text files, XML or other structured files, or other proprietary formats. Some embodiments of the invention format the semantic content according to an XML language called VirtualModem™ Markup Language (VMML) to represent the semantic features of HTML pages and TVML presentations. The rendered content 160 and semantic content can be stored in a traditional cache, a database, a file system or other storage media. The underlying file system can be used to store the content in a directory and file hierarchy that represents the rendered cache 201.

The rendered content 160 stored 320 in the rendered cache 201 can include images, audio, text, full motion video, animations, etc. The content is stored in the rendered cache 201 regardless of its format [i.e. the rendered cache

201 can store binary large objects (blobs) or format-independent objects]. The format in which the semantic content is stored is independent of the rendered cache 201 mechanism. The content browsers and other client applications that access the rendered content 160 stored in the rendered cache 201 include processing resources adapted to recognize the format and interpret the semantic content appropriately.

According to some embodiments of the invention, content browsers and other client applications include processing resources to recognize and play 170 the rendered content 160 after the corresponding format-independent objects are retrieved 260 from the rendered cache 201. Some embodiments of the invention, including various VirtualModem™ presentation systems, can render HTML pages into a proprietary image format, called a fat macroblock (FMB), that is suitable for display on televisions. FMB's are described in greater detail by United States patent application serial number 09/287,235, entitled "System and Methods for Preparing Multimedia Data Using Digital Video Data Compression", filed April 6, 1999, having inventors Antoine Boucher, Paul E. McRae, and Tong Qiu, the entire contents of which are hereby incorporated herein by reference as if fully set forth herein.

In the case where the content is not missing but is outdated, the entire content can be retrieved 260, or just the outdated portions can be retrieved. By retrieving 260 only outdated portions some savings can be gained in the rendering 150 step by eliminating the need for a full rendering. For example, perhaps only an animated image on an HTML page has changed in the requested content. The rendering system can detect this situation and render 150 only the new animation rather than the entire page.

Once the needed portion of the request content has been retrieved 260, the content is rendered 150 before it is played 170. The retrieved content is handed to a rendering system that typically performs the following actions:

1. Laying out 140 of the content according to the appropriate rules (e.g., HTML rules)

2. Rendering 150 the content according to the rendering instructions

140A, thereby producing presentation data (e.g., for an MPEG image formatted as an FMB- or set of images for HTML frames) that represent the fully rendered representation of the content (e.g., the HTML page). The page may also have other graphical elements created for such things as animated GIFs.

3. Generating 315 a semantic representation 140B of the semantic features. Generally, the layout engine or the render engine creates the semantic representation 140B from the layout 140 or play instructions. For an HTML page, the semantic representation 140B can include the location, size, shape, and target of all HTML anchors (links to other HTML pages), the timing, location, and size of animated GIF frames, the size and relative location of HTML frames, information on HTML forms that can be accessed from the page, and HTML meta values.

4. Storing 320 the rendered content 160 [e.g., MPEG image(s)] in the rendered cache 201 using an appropriate index, e.g., a URI. The semantic content is also stored 320 in the rendered cache 201 using an appropriate index. In some embodiments, the semantic content can be stored 320 in an XML-based format so that it can be easily parsed and restored (e.g., in step 240) in the future. After the rendering system is finished, the rendered content 160 can be provided to the user by simply reading and restoring 240 and playing 170 the content.

The "format" of the semantic representation 140B is determined by the engine that generates the semantic representation (e.g. Netscape Communicator and Microsoft ® Internet Explorer use the DOM). This internal semantic representation 140B is then stored as a physical entity (semantic content) in the rendered cache 201. The format of semantic content is adapted for the browser engine that reads the semantic content for play 170. The format of the semantic content is sufficiently detailed for the browser engine to create its own semantic representation 140B. The semantic representation 140B in the browser engine can be the same internal format that the layout/render engine uses or the semantic representation in browser can have a different format.

As shown in Figure 4, when a request is received for content already in the rendered cache 201 the rendering system process can be skipped entirely. The following simple steps are all that is involved to play 170 content already in the rendered cache 201.

1. Read the semantic content and the rendered content 160, and restore the semantic representation 140B from the semantic content stored in the rendered cache 201, e.g., the VMML description.

2. Play 170 the rendered content 160 on the user's screen according to this semantic representation 140B.

Some multimedia content, such as an HTML web page, does not fit entirely on a user's screen at once. For such partial page displays, the browser can use the semantic representation 140B to determine which portion of the page should be displayed, and for some embodiments which subset of the hypertext links are selectable on the page portion. An example of this scrolling is described below in the "HTML Page with Layout" example below.

Retrieving Content from the Rendered Cache

When a content browser, or other client application, requests a target index, such as a URI, the rendered cache 201 mechanism first looks in the rendered cache for a rendered representation of the content. The caching mechanism provides a means to search and retrieve this content based on the content's indexing indication. Examples of cache retrieval mechanisms include database queries, simple index files, file system directory structures, or traditional browser caches.

If the rendered content 160 can be found in the rendered cache 201, the content will be displayed very quickly. The semantic representation 140B of the rendered content 160 will be restored using the semantic content stored in the rendered cache 201 (i.e. the semantic features need not be computed again before the rendered content is played). For example, some embodiments restore the semantic representation 140B of a rendered HTML page by reading the VMML formatted semantic content.

If the content browser cannot locate a rendered representation of the multimedia content in the rendered cache 201, or the browser determines that the content is out of date, then the content can be retrieved 260 (either from a traditional cache 110, from the Internet 105, or from another content source) and rendered 150. The retrieval 260 and rendering 150 results in at least one new rendered cache 201 entry that can be used the next time the multimedia content is accessed.

A system that uses a rendered cache 201 will, after determining that no rendered representation is in the cache, perform the same steps as described above. That is, the HTML source will be read and the page laid out 140. The resulting rendering instructions 140A are followed but rather than displaying the page (or, alternatively, in addition to displaying) the page the rendering will be stored as a graphical image in the rendered cache 201. The semantic content (describing the location, size, and target URI of the single hyperlink on the image) is also stored in the rendered cache 201. The next time and every subsequent time the browser receives a request to view this URI, the browser simply reads the semantic content and the rendered content 160, restores the semantic representation 140B, and displays the rendered content. Thus, the use of the rendered cache 201 saves the cost of processing for layout 140, generation 315 of the semantic representation 140B, and rendering 150B. For more complicated HTML pages this savings can be substantial.

Examples

Specific embodiments of the invention are further described by the following, non-limiting examples which will serve to illustrate in some detail various features of significance. The examples are intended merely to facilitate an understanding of ways in which the invention may be practiced and to further enable those of skill in the art to practice the invention. Accordingly, the examples should not be construed as limiting the scope of the invention.

An HTML Page with Layout using a Distributed Server-Based Content

System

Some embodiments of the invention provide storage 320, retrieval 260 and/or play 170 of HTML pages. One embodiment of the invention is represented by the Toronto Stock Exchange (TSE) HTML homepage illustrated by Figures 5A and 5B.

For this embodiment, the "content browser" can be broken up into a distributed server-based content preparation and viewing system. The viewing system can include a display device, e.g., a television, and a digital set-top box (such as a General Instruments DCT-2000).

For some HTML page embodiments, the set-top box has neither the processing nor the storage resources needed to render 150 or cache content. The set-top box typically does have the capability to decode and play MPEG images and Dolby AC-3 audio, and some limited graphics capabilities in order to do text and simple graphical overlays. For these embodiments, all access to rendering 150 processing resources and content stored in the rendered cache 201 is done at the server. These embodiments are described in greater detail in the "System" section below.

In other embodiments, the set-top box, or other addressable processing equipment, can have processing resources and storage medium capable of rendering 150 and caching the content. In response to the server-based system receiving a request to view some content with the URI http://www.tse.com/ and determining that the content is either not in the rendered cache or is outdated, the server system browser requests retrieval of the TSE web page and any graphical elements the TSE web page references.

Once the web page and graphics have been retrieved 260 (either from a traditional cache 110 or from the Internet 105) the browser requests that the content be laid out 140 and rendered 150. The rendering system creates an MPEG representation (in FMB format) of the rendered web page. Because MPEG is the only image format the GI DCT-2000 recognizes, we use MPEG in this example. The rendering system can also generate other FMB files representing animated GIF frames, if animated GIFs were referenced in the

HTML page. The rendering system also creates a semantic representation 140B of the page including the location, shape, size, and target of all hyperlinks; location, size, and timing of animated GIF frames; HTML form information; and HTML meta information.

The FMB files are stored 320 in the rendered cache 201 using the URI of the HTML page ("www.tse.com") as an index. The semantic content is also stored 320 in the rendered cache 201 using the URI as an index. The semantic content is stored in an XML format called VMML. For distributed systems embodiments, e.g., the VirtualModem™ system, the internal semantic representation 140B for the layout/render engine is different than the semantic representation 140B for the browser engine (although these semantic representations 140B are conceptually equal). The stored semantic content (in the form of VMML for VirtualModemTM) is detailed enough to allow for "information transfer" so that two different semantic representations 140B can be used.

Once the rendered content 160 (FMBs) and semantic content (VMML) are stored 320 in the rendered cache 201, the browser can then read and restore the semantic representation 140B based on the VMML file. Using this semantic content the web page can be displayed.

The first screen capture (Fig. 5A) of the TSE homepage shows the top portion of the page. The rectangular highlight box 510 in the top left corner indicates that the user can select the first hyperlink for viewing. Users can press arrow keys on their remote control to move from one link to another link on the page. The browser provides enough information for the set-top box to draw the highlight box 510 and to navigate the page from link to link using the arrow keys.

Eventually the user may scroll past the bottom of the screen. The set-top will then inform the server-based browser that a scroll is required and the browser will then determine from the semantic content which new portion of the rendered MPEG should be visible and which new subset of the hyperlinks is now selectable.

The second screen capture (Fig. 5B) illustrates the TSE homepage after a scroll down. The user can continue to view the same page scrolling around and viewing the content in the fashion described above. However, once a link is selected the browser is informed of the corresponding new URI request and the retrieval 260 (or read and restore 240) process are initiated again after the browser receives a play request 120.

An HTML Page with Layout using a Self-Contained Content System

Some embodiments do not use the distributed server-based content preparation and viewing system described in the above example (HTML Page with Layout). Instead, these embodiments are self-contained content systems with layout 140, rendering 150, and play 170 processes all combined in a single computer program. Netscape Communicator and Microsoft ® Internet Explorer are examples of such embodiments.

Such "traditional" browsers can also use the invention to reduce retrieval 260 and playing 170 time, and decrease processor usage. Traditional web browsers have long used caching technologies to minimize the need to use slower content retrieval 260 methods such as network access. These browsers store the original retrieved content in a cache database. When a request to view content is received, the browser searches the traditional cache 110. If the content not in the cache then the browser retrieves the content from an alternate source (such as the Internet 105). Visiting web sites that reside in a rendered cache 201 results in almost instantaneous display of the web site content rather than the usual delay (due to the cost of layout, rendering and creation of semantic context) that is normally seen.

Whether or not the content was found in the traditional cache 110, the content is then read and laid out 140 according to the rules of HTML. Laying out 140 produces rendering instructions 140A and a semantic representation 140B of the content. The page is then rendered 150 to a graphical format (typically a bitmap) and played 170 according to the semantic representation 140B. These steps are performed each and every time the content is requested.

For the self-contained embodiments of the invention, when a request for content is received the browser will search in the rendered cache 201 to determine whether a rendered representation of the content is available. If the content is not in the rendered cache 201, or if the rendered content is found to be outdated, then the content must be requested from an alternate source (such as the Internet 105, or a traditional cache 110). Once the content is received it will go through the same layout 140, rendering 150, and generation 315 of the semantic representation 140B steps as these browsers do now.

The difference is that once the rendering 150 and generation 315 of the semantic representation 140B is complete the rendered content 160 and the semantic representation 140B are stored in the rendered cache 201.

Once the content is stored in the rendered cache 201, then each time the browser receives a request for this content, the browser simply reads and restores 240 the semantic representation 140B and plays 170 the rendered content 160 according to this semantic representation. The format of the rendered content and semantic representation are entirely up to the browser. It is recommended that the rendered content be stored in a "native format". That is, a format that the browser can immediately recognize and does not have to convert to a recognized format. It is also recommended that the format for the semantic representation 140B be rich enough to cover all the various semantic elements that HTML can describe. VMML is a good example of such a format. For self-contained systems the format of the internal semantic representation 140B is likely to be the same for both the layout 140/render 150 and browser portions of the program.

Another related embodiment that could benefit from the invention is what is commonly referred to as a "web proxy". A web proxy is a computer program that retrieves content on behalf of content browsers. Various embodiments of the invention enable the web proxy to only retrieve 260 content from the Internet 105 for the first request, while all future requests for the content from browsers using the proxy use the locally cached version.

Note that in either the distributed or self-contained scenarios, the task of converting to and from the stored semantic content format is up to the relevant engines (layout 140/render 150 engine for storing and browser engine for retrieval 260.

An alternate scenario could involve the layout/render engine transferring the semantic representation (through some communications medium) to a "rendered cache server" that converts the representation into semantic content. This server would also receive request to retrieve content from the cache and would read the semantic content, convert it to an appropriate internal representation and then transfer this representation. In this case the task of converting to and from semantic content is entirely up to the "rendered cache server". In

practice, this approach is less flexible than alternative approaches.

In the case where the web proxy and the content browsers all have access to the same storage or have access to a fast internal communications network the web proxy could perform the layout 140, rendering 150, and generation 315 of semantic representation 140B steps on behalf of the content browsers. In such a scenario, when a content browser receives a request for content, the content browser can either look directly in the rendered cache 201 or query the web proxy for the rendered content 160. The browser can then simply read and restore 240 the semantic content and display the rendered content 160 accordingly. This use of the web proxy allow for the use of very small and efficient web browser implementations since all the resources for layout 140, rendering 150, and generation 315 of the semantic representation 140B are external of the browser.

An intelligent web proxy can pre-render the content that it downloads in order to offset the rendering cost in browsers. This approach is especially beneficial in situations where client computing resources are limited. A key application of this approach is in the emerging market of set-top devices and other network computers. These devices typically have tightly constrained resources and do not presently provide true web browsing. The use of a rendered cache proxy would offload the process of layout 140, rendering 150, and generation of the semantic representation 140B.

In some embodiments, word processing programs can store 320 rendered documents in a rendered cache 201 for faster loading and previewing. Using the rendered cache 201 for storing 320 word processing documents also enables programs other than the word processor to preview the content without using proprietary plug-ins or libraries.

Multimedia Content with Play Instructions

Some embodiments of the invention provide storage 320, retrieval 260 and/or play 170 of multimedia content. The multimedia content can include images, audio, text, graphics, and full motion video, all of which can be timed to play at different moments. This multimedia content can have a means of referencing other multimedia content in a manner similar to HTML hyperlinks. Some embodiments of the invention, including the VirtualModem^{TM} system from Interactive Channel, use an XML language called TVML to represent the play instructions of a multimedia presentation. TVML can include markup to represent the following play instructions of the multimedia content:

1. Timing of multimedia content playing;
2. Order of multimedia content playing;
3. Size and location of multimedia content; and
4. Location, size, shape, and target URI (or other index) of hyperlinks.

One embodiment of the invention is represented by the News Menu TVML presentation illustrated by Figs. 6A through 6D. Fig. 6A illustrates a timeline representing how the News Menu TVML presentation should be played. Figs. 6B through 6D show the images that make up the News Menu TVML presentation. As in the previously described embodiment (HTML with layout) the "content browser" can be broken up into a distributed server-based content preparation and viewing system.

The server-based system can receive a request to view some content with the URI http://www.virtualmodem.com/news.tvml and then determined that the content is either not in the rendered cache 201 or is outdated. The browser can respond to this circumstance by submitting a request to retrieve the TVML presentation and any multimedia elements referenced by the presentation. Once the presentation and its multimedia elements have been retrieved 260 (either from a traditional cache 110 or from the Internet 105), the browser requests that the content be rendered 150. In this case, layout 140 is unnecessary and the rendering 150 can be limited to converting the multimedia content into a format that the set-top recognizes. In the case of the GI DCT-2000 images and full motion video are converted to MPEG formatted data and audio is converted to Dolby AC-3 formatted data.

The rendering system can also generate 315 a semantic representation 140B of the page from the TVML play instructions. The semantic representation 140B can include context such as the relative play times and order of the multimedia content; the location, shape, size, and target of all hyperlinks; and TVML meta information. The rendered content 160 can be stored 320 in the rendered cache 201 using the URI of the presentation ("www.virtualmodem.com/news.tvml") as an index. The appropriately formatted semantic content based on the semantic representation 140B is also stored in the rendered cache 201 using the URI as an index. For some embodiments of the invention, the semantic content is stored in a VMML format.

Once the rendered content 160 and semantic content (VMML) are stored in the rendered cache 201, the browser can read and restore 240 the semantic representation 140B from the VMML file in which the semantic content is disposed. Using this semantic representation 140B the presentation can be displayed.

Figure 6A shows the start time and duration that each image of the presentation should be played, and illustrates the start time and duration of the accompanying audio. The presentation plays 170 from t_0 to t_3.

Figures 6B through 6D show each of the images used in the for the News Menu TVML presentation. The first image 650 of the presentation, shown in Fig. 6B, includes a single circle with a small diameter around the top of the transmitter to indicate that a signal is being sent from a transmitter. As shown by the first time line 610, the first image 650 is shown from t_0 to t_1.

The second image 660 of the presentation, shown in FIG. 6C, includes three circles around the top of the transmitter to indicate that the signal will be received by the user sooner than when the first image 650 was displayed. As shown by the second time line 620, the second image 660 is shown from t_1 to t_2.

The third image 670 of the presentation, shown in FIG. 6D, includes a first hyperlink that is enclosed by a rectangular highlight box 510 to indicate that the first hyperlink "World News Update" is presently available for selection. As shown by the second time line 630, the third image 660 is shown from t_2 to t_3. A user can press arrow keys disposed on the user's remote control devices to move from link to link in the third image 670. The browser provides enough information for the set-top box to draw this rectangle and to navigate using the arrow keys from link to link. If the multimedia content is larger than the physical screen then it becomes possible to scroll in the same manner as described in the "HTML with layout" example. As shown by the fourth time 640, the accompanying audio plays 170 for the entire duration of the News Menu TVML presentation.

In the News Menu TVML presentation example only the third image 670 of the presentation contains hyperlinks. However, in other embodiments of the invention, any of the earlier images can also contain hyperlinks. The browser can update the client (set-top box or other addressable processing equipment) whenever the semantic representation 140B (e.g. hyperlink information or image display duration) changes.

The presentation can play 170 until all multimedia objects have been played. The user can continue to view the last image of the presentation in the same manner as for HTML pages. The user can also manipulate the remote control VCR functions to rewind, fast-forward, or pause the presentation. However, once a hyperlink is selected the browser will be informed of the new URI request and the content retrieval process will start again with a request for content.

Systems for Storing, Retrieving and Playing Multimedia Content

Some embodiments of the invention include systems for storing 320, retrieving 260 and playing 170 multimedia content using a rendered cache 201. Listed below are the key elements of a system that can implement various embodiments of the invention. Previous descriptions and examples mentioned in the "Distributed Server-Based Content System" section have illustrated the use of the invention in a distributed server-based system. In such a system the various complimentary components, such as those listed below, are typically found in separately running processors that can reside in a single computer or in multiple connected computers. Some embodiments, such as the VirtualModem^{TM} system can include the following components:

Web crawler processing resources adapted to access multimedia content from source data storage. The multimedia data can include HTML and TVML content. The source data storage can include at least one of the Internet 105 and a web proxy cache.

Rendering processing resources adapted to generate semantic representation 140B of, and render 150 multimedia data, and can format the semantic representation as semantic content. In some embodiments, a rendering program can also be adapted to layout 140 the multimedia data.

Multimedia playing processing resources, such as an audio/video terminal server (AVTS), adapted to play multimedia content. Such play can include displaying images and playing audio and full motion video. Some embodiments of an AVTS are described in greater detail in United States patent application serial number 09/255,052, entitled "System and Method for Interactive Distribution of Selectable Presentations," filed February 22, 1999, and having inventors: Antoine Boucher, James Lee Fischer, and Allan E. Lodberg, the entire contents of which are hereby incorporated herein by reference as if fully set forth herein.

Browser processing resources adapted to interpret the semantic content and control when and how the multimedia content should be played. The browser processing resources can act as the "control center" for the entire process. The browser processing resources can communicate with the web crawler, rendering, and the multimedia playing processing resources and coordinate the interactions of each of these.

A second group of embodiments discussed above in the "Self-Contained Content System" section contain all the required rendered cache 201 complimentary components in a single program. Netscape Communicator and Microsoft ® Internet Explorer are both examples of such single programs. Both of these browsers include programs adapted to retrieve 260 content from the Internet 105 or a proxy. Netscape Communicator and Microsoft ® Internet Explorer also have layout 140, rendering 150, and semantic representation 140B generating capabilities. They both have the ability to display the rendered content 160 to a computer monitor display and they both can interpret the semantic representation 140B.

FIG. 7 illustrates the components and features configured in a system for accessing multimedia content using a rendered cache 700 representing one embodiment of the invention. The system for accessing multimedia content using a rendered cache 700 includes the components and features described below, including: access to source content 710, at least one layout engine 720, at least one render engine 730, a rendered cache 201, at least one multimedia play engine 750, at least one browser engine 760, and a display 770. These components can be combined together to form one or more computer programs that implement the storing 320, retrieving 260 and playing 170 methods described above.

Source content is content that is not yet rendered. The source content can include HTML, XML, images, audio, text, and full motion video. Access to source content 710 can be through an Intranet, the Internet 105, a web proxy, or on local storage. Connections adapted to provide such access can be through any carrier capable of providing sufficient bandwidth for practical retrieval 260 the content, such as: digital subscriber line (DSL), cable modem, T-1, T-2, T-3, OC-1 through OC-256, fiber distributed data interface (FDDI), E1 through E5, Ethernet, fast Ethernet, and Gigabit Ethernet. Access to source content 710 can also include processing resources adapted to use standard Internet protocols such as TCP/IP and HTTP, and to read files from a file system. The component providing access to source content 710 includes processing resources for retrieving the source content, such as the content fetch 715 resources shown in FIG. 7.

The system for accessing multimedia content using a rendered cache 700 can include layout processing resources, such as a layout engine 720, adapted to derive rendering instructions 140A from a content definition (e.g., HTML). The layout engine 720 can also derive a semantic representation 140B of the features of the content from the layout 140, or from the play 170 instructions. Netscape Communicator and Microsoft ® Internet Explorer both contain processing resources to perform HTML layout 140 as part of their overall functionality. Stand-alone layout engines 720 include Spyglass Device Mosaic, NGLayout from Mozilla, and Chimera. In some embodiments, processing resources other than the layout processing resources can be adapted to generate 315 the semantic representation 140B from play 170 instructions.

The system for accessing multimedia content using a rendered cache 700 can include rendering processing resources, such as a render engine 730, adapted to create a graphical representation of content that has been laid out 140 by the layout engine 720. The rendering engine 730 can also have the capability of converting content that does not require layout 140 into a form that is ready for rapid play 170.

Many layout engines 720 also include a render engine 730. Systems that have layout engines 720 that do not include a rendering 150 capability have a separate render engine 730 and typically specify the interface that a rendering engine must have (e.g., this is how Spyglass Device Mosaic works). Both Netscape Communicator and Microsoft ® Internet Explorer include rendering engines as part of their overall functionality. Both of these browsers render 150 the content into a bitmap that can be displayed on a computer monitor display

770. Some embodiments, such as the VirtualModem™ system, use their own custom render engine 730 that renders the content to MPEG files stored in FMB format.

The rendered cache 201 provides access to an indexed storage mechanism. The rendered cache 201 stores both the rendered content 160 and the semantic content so that these data objects can be easily retrieved 260 at a later time. The rendered cache 201 includes as indexing mechanism that can take a variety of forms including database queries, index files, file system directories.

The format of the rendered content 160 is independent of the storage mechanism. A format that requires little or no conversion at play 170 time (i.e. a "native format") provides greater time and processing savings using the rendered cache 201.

The format of the semantic content is also independent of the storage mechanism. The semantic content format used in the system for accessing multimedia content using a rendered cache 700 that fully captures all the semantic features of the rendered content 160 provides enhanced play results. The semantic content format can avoid unneeded complexity to ensure that the processing and time required to restore the semantic representation 140B are less than that required to layout 140 and re-render the content.

The system for accessing multimedia content using a rendered cache 700 includes multimedia play processing resources, such as a multimedia play engine 750, adapted to play the rendered content 160 on a display 770 device. The multimedia play engine 750 can read the rendered content 160 directly from the rendered cache 201 indexed storage mechanism, read the rendered content from memory, or otherwise receive the rendered content from an external source. Netscape Communicator and Microsoft ® Internet Explorer both contain, as part of their overall functionality, processing resources to display multimedia content to a computer display 770. Some embodiments, including the VirtualModem™ system, include a separate program that is part of the overall distributed system, called the AVTS, that is adapted to play multimedia content to APEs.

The system for accessing multimedia content using a rendered cache 700 also includes a browser engine 760 adapted to interpret the semantic representation 140B of the rendered content 160 being played 170. The browser engine 760 can read the semantic content directly from the rendered cache 201 indexed storage mechanism, or interpret the rendered content 160 from memory, or otherwise receive the semantic content from an external source.

The browser engine 760 can be adapted to interpret the semantic features from the semantic content. In some embodiments, the browser engine 760 is adapted to control navigation of hyperlinks (i.e. determining from user input which content should be displayed next). The browser engine 760 also can determine which portions of the rendered content 160 should be played 170, and which corresponding portions of the semantic representation 140B are active (e.g., when scrolling an image).

The browser engine 760 can be included in commercially available software such as Netscape Communicator, Microsoft ® Internet Explorer, or any other browser engine that is adapted to perform the functions described above. Netscape Communicator and Microsoft ® Internet Explorer both contain, as a part of their functionality, processing resources adapted to interpret a semantic representation 140B [or Document Object Model (DOM) as both call it]. Both of these browsers use the DOM to determine which links are currently visible (and which others are scrolled off the screen), animated GIF timing and location, information about HTML forms and other HTML features. Some embodiments, such as the VirtualModem™ system, include a browser program that coordinates the retrieving 260 of content, layout 140 and rendering 150 of content, and playing 170 of rendered content. These browser embodiments can also contain processing resources for reading semantic content from the rendered cache 201 and restoring the semantic representation 140B.

The above engines (layout, render, play, and browser) are all at least loosely coupled. That is, they need not be part of the same program but there needs to be some form of communication between them all. This communication can take a variety of forms including inter-process communication (such as shared memory, pipes, or messaging protocols), or shared files. Some embodiments, such as the VirtualModem™ system, use a communications protocol built on a user data protocol (UDP) to communicate between the various engines. Netscape Communicator and Microsoft ® Internet Explorer include all the engine components in the same program.

There is no requirement that any of the above system components be directly tied together (i.e. included in the same program). However, there are advantages to tightly coupling certain components. For example, it is more efficient to couple the layout engine 720 and the render engine 730 in the same program. In such a scenario the rendering instructions 140A resulting from layout 140 process can be used directly by the rendering engine 730 component. If the layout engine 720 and the render engine are separate programs, then some intermediate form of rendering instructions (e.g. either a file or data passed over a network) would have to be used.

Only the component responsible for accessing source content 710 needs to include processing resources to access the communications carrier and the underlying communications protocol. It is not required that the other engine components have these processing resources.

The layout engine 720 and the render engine 730 has access to the rendered cache 201 storage mechanism since they read the rendered content 160 and the semantic content.

The multimedia play engine 750 has access to at least the rendered content 160 portion of the rendered cache 201 storage mechanism. The browser engine 760 has access to at least the semantic content portion of the rendered cache 201 storage mechanism. Both the multimedia play engine 750 and the browser engine 760 can have full access to the entire rendered cache 201 storage mechanism but at minimum they have access to their respective content.

Splitting access to the rendered content 160 and the semantic content allows for efficient distribution of the multimedia play engine 750 and browser engine 760.

Formatting the Semantic Content

Some embodiments of the invention use an eXtensible markup language (XML) language to format and store 320 semantic content in the rendered cache 201. Embodiments including the VirtualModem™ system use a markup language called VMML to format and store 320 semantic content in the rendered cache 201.

VMML contains elements to describe the semantic features of both HTML and TVML. TVML is another XML language originally based on synchronized multimedia integration language (SMIL) from the World Wide Web Consortium at http://www.w3.org/. The descriptive elements include:

1. Multimedia elements - The <img>, <audio>, <video>, and <text> elements are used to describe fully rendered multimedia objects. The <screen> element is used to describe fully rendered HTML. Each of these elements can include an optional start time using the "begin" attribute.

2. Aggregation elements - The <par> and <seq> elements are used to describe how the multimedia elements are played. Elements inside a <par> are played in parallel. The start times of multimedia elements in a <par> are relative to the beginning of the <par>. Elements inside a <seq> are played sequentially. The start times of multimedia elements in a <seq> are relative to the end of the previous element. Both the <par> and <seq> elements can define optional start times using the "begin" attribute.

For example, the following <par> element contains an <audio> and two <img> elements which are played in parallel (i.e. at the same time). The display of the second image is delayed by 5 seconds.

```
<par>
        <audio src="voice-over.ac3"/>
        <img  src="first-screen.fmb"/>
        <img  src="second-screen.fmb" begin="5.0s"/>
</par>
```

3. HTML elements - The <screen> element is used as a container for all the semantic information concerning a rendered HTML page. Elements allowed in a <screen> element include:

<frame> - contains attributes for defining the FMB (the rendered frame), size, and location relative to other frames of the HTML page;

<anchor> - each <frame> element can contain a list of <anchor> elements which describe the location, size, shape, and target of HTML hyperlinks;

<form> - each frame can contain form elements which fully describe HTML forms;

<animation> - <frame> elements can contain animation elements that describe the timing, size and location of animated GIFs.

4. Non-display elements - The <title> and <meta> elements describe non-audiovisual features of the content. Examples of <meta> information include HTML refreshes, and expire metas.

5. Anchors - Information about non-HTML hyperlinks is also described in VMML <anchor> elements.

6. Applets - The <applet> element instructs the browser to run other applications.

VMML contains other minor elements and a wide variety of attributes but the above list describes the major features. VMML is capable of describing all the various features of TVML and HTML in sufficient detail that the semantic representation 140B can be reconstructed after reading the semantic content from the rendered cache 201. The reconstruction of the semantic representation 140B includes simple tokenization (i.e. text parsing) using freely available tools such as sgml-lex (available from http://www.w3.org/). The parsing process is much faster and uses far fewer processor resources than the processes of layout 140 and rendering 150.

The term coupled, as used herein, is defined as connected, although not necessarily directly, and not necessarily mechanically. The term substantially, as used herein, is defined as approximately (e.g., preferably within 10% of, more preferably within 1% of, most preferably within 0.1% of).

Advantages of the Invention

A rendered cache 201 coupled with multimedia content render, play, and browser processing resources, representing an embodiment of the invention, can be cost effective and advantageous for at least the following reasons. The rendered cache 201 enables the play 170 of multimedia content in less time and using less data processing because the steps of layout and rendering are eliminated.

All the disclosed embodiments of the invention described herein can be realized and practiced without undue experimentation. Although the best mode of carrying out the invention contemplated by the inventors is disclosed above, practice of the invention is not limited thereto. Accordingly, it will be appreciated by those skilled in the art that the invention may be practiced otherwise than as specifically described herein.

For example, although the rendered cache 201 described herein can be a physically separate module, it will be manifest that the rendered cache 201 can be integrated into the apparatus with which it is associated. Furthermore, all the disclosed elements and features of each disclosed embodiment can be combined with, or substituted for, the disclosed elements and features of every other disclosed embodiment except where such elements or features are mutually exclusive.

It will be manifest that various additions, modifications and rearrangements of the features of the invention may be made without deviating from the spirit and scope of the underlying inventive concept. It is intended that the scope of the invention as defined by the appended claims and their equivalents cover all such additions, modifications, and rearrangements. The appended claims are not to be interpreted as including means-plus-function limitations, unless such a limitation is explicitly recited in a given claim using the phrase "means-for." Expedient embodiments of the invention are differentiated by the appended sub-claims.

CLAIMS

What is claimed is:

1. A method, implemented in at least one computer, for storing multimedia data, comprising:

detecting multimedia content, the multimedia content includes play instructions and at least one multimedia element, the at least one multimedia element includes at least one of graphical images, audio, text, and full motion video;

generating a semantic representation of a rendered representation of the multimedia content from the play instructions, the play instructions include at least one of timing of the multimedia content and ordering of the multimedia content, the semantic representation describes at least one of: characteristics of the rendered representation, and relationships between different multimedia elements disposed in the rendered representation;

storing the rendered representation of the multimedia content in a storage medium; and

storing data corresponding to the semantic representation in the storage medium.

2. The method for storing multimedia data of claim 1, wherein the semantic representation includes at least one of:

a location of a hyperlink;

a size of the hyperlink;

a shape of the hyperlink;

a target index of the hyperlink;

a size of a portion of the multimedia content disposed in a rendered image;

a relative location of the portion of the multimedia content disposed in a rendered image;

sizes of animated formatted graphics;

locations of the animated formatted graphics;
timing of the animated formatted graphics;
browser commands for addressable processing equipment; and
data corresponding to the play instructions.

3. The method for storing multimedia data of claim 1, wherein:
one of the at least one computer includes rendering processing resources adapted to generate the semantic representation; and
storing the data corresponding to the semantic representation includes:
formatting the semantic representation into semantic content to enable rapid restoring of the semantic representation by a content browser; and
storing the semantic content.

4. The method for storing multimedia data of claim 3, wherein:
storing the semantic content includes the rendering processing resources transferring a semantic content file to the storage medium;
storing the rendered representation includes the rendering processing resources transferring at least one rendered representation file to the storage medium; and
the method includes indexing the semantic content file and the at least one rendered representation file based on at least one universal resource identifier, to form at least one indexed file.

5. The method for storing multimedia data of claim 4, including arranging the indexed files to enable a client application to retrieve the rendered representation and the semantic content, the client application includes the content browser.

6. The method for storing multimedia data of claim 1, including converting the multimedia content into the rendered representation of the multimedia content.

7. A method, implemented in at least one computer, for storing multimedia data, comprising:

detecting multimedia content including layout instructions, the multimedia content includes at least one multimedia element, the at least one multimedia element includes at least one of graphical images, audio, text, and full motion video;

laying out the multimedia content according to the layout instructions to form rendering instructions and a semantic representation of a rendered representation of the multimedia content, the semantic representation describes at least one of: characteristics of the rendered representation of the multimedia content, and relationships between different multimedia elements disposed in the rendered representation of the multimedia content;

rendering the multimedia content according to the rendering instructions to produce the rendered representation;

storing the rendered representation in a storage medium; and

storing data corresponding to the semantic representation in the storage medium.

8. The method for storing multimedia data of claim 7, including:

detecting play instructions, and

wherein laying out includes incorporating the play instructions into the semantic representation, and the play instructions include at least one of timing of the multimedia content and ordering of the multimedia content.

9. A method, implemented in at least one computer, for retrieving multimedia data, comprising:

processing resources of a first computer of the at least one computer detecting a request for requested multimedia content;

processing resources coupled with the first computer determining whether data corresponding to the requested multimedia content is disposed in a storage medium, the storage medium is coupled with the first computer, the storage medium includes rendered representations of multimedia content and semantic content, the semantic content includes data corresponding to semantic representations derived from one of: play instructions for the rendered content, and layout of the multimedia content, the semantic representations describe at least one of: characteristics of the rendered representations, and relationships between different multimedia elements disposed in the rendered representations; and

responding to a determination that data corresponding to the requested multimedia content are disposed in the storage medium by:

retrieving a rendered representation of the requested multimedia content; and

retrieving semantic content corresponding to the requested multimedia content.

10. The method for retrieving multimedia data of claim 9, including restoring the semantic representation for the requested multimedia content using the semantic content corresponding to the requested multimedia content.

11. The method for retrieving multimedia data of claim 9, wherein determining whether data corresponding to the requested multimedia content are disposed in the storage medium includes searching the storage medium using a retrieval mechanism adapted to search and retrieve content based on an index corresponding to the requested multimedia content.

12. The method for retrieving multimedia data of claim 11, wherein a semantic representation corresponding to the requested multimedia content includes play instructions for the requested multimedia content, and the method includes playing the requested multimedia content according to the play instructions.

13. The method for retrieving multimedia data of claim 11,

including: responding to a determination that data corresponding to the requested multimedia content are not disposed in the storage medium by:

storing the data corresponding to the requested multimedia content in the storage medium;

retrieving the rendered representation of the requested multimedia content; and

retrieving the semantic content corresponding to requested multimedia content.

14. The method for retrieving multimedia data of claim 13 including restoring a semantic representation for the rendered representation of the requested multimedia content using the semantic content corresponding to requested multimedia content.

15. The method for retrieving multimedia data of claim 9 including responding to a determination that data corresponding to the requested multimedia content is not disposed in the storage medium by:

storing data corresponding to the requested multimedia content in the storage medium;

retrieving the rendered representation of the requested multimedia content; and

retrieving the semantic content corresponding to requested multimedia content.

16. The method for retrieving multimedia data of claim 10, wherein:

data disposed in the storage medium includes at least one file;

the at least on file includes the rendered representation of the requested multimedia content, and the semantic content corresponding to the requested multimedia content;

restoring the semantic representation is performed by a client application using the semantic content corresponding to requested multimedia content; and the method includes:

a client computer requesting the requested multimedia content; and

the client application recognizing and playing the requested multimedia content from at least a portion of the at least one file.

17. The method for retrieving multimedia data of claim 9, wherein responsive to determining that data corresponding to the requested multimedia content are disposed in the storage medium, determining whether the data corresponding to requested multimedia content disposed in the storage medium require updating.

18. The method for retrieving multimedia data of claim 9, including, prior to retrieving the rendered representation of the requested multimedia content:

processing resources coupled with the first computer determining whether the data corresponding to the requested multimedia content disposed in the storage medium require updating:

responsive to a determination that the data corresponding to the requested multimedia content disposed in the storage medium require updating:

storing an updated version of the data corresponding to the requested multimedia content in the storage medium;

retrieving at least a portion of an updated version of the rendered representation of the requested multimedia content;

retrieving at least a portion of an updated version of the semantic content corresponding to the requested multimedia content; and

restoring a semantic representation for the requested multimedia content using the at least a portion of the updated version of the semantic content.

19. The method for retrieving multimedia data of claim 18, wherein

determining whether the data corresponding to the requested multimedia content disposed in the storage medium require updating includes at least one of:

comparing an expiry date tag for the data corresponding to the requested multimedia content disposed in the storage medium with a date corresponding to the updated version of the data corresponding to the requested multimedia content; and

comparing the data corresponding to the requested multimedia content disposed in the storage medium with a version of data corresponding to the requested multimedia content disposed in a different storage medium.

20. The method for retrieving multimedia data of claim 9, including, prior to retrieving the rendered representation of the requested multimedia content:

processing resources coupled with the first computer determining whether the data corresponding to the requested multimedia content disposed in the storage medium require updating; and

responsive to a determination that the data corresponding to the requested multimedia content disposed in the storage medium require updating:

storing an updated version of the data corresponding to the requested multimedia content in the storage medium including an updated version of the rendered representation of the requested multimedia content, and an updated version of the semantic content corresponding to the updated version of the rendered representation;

retrieving the updated version of the rendered representation of the requested multimedia content;

retrieving the updated version of the semantic content corresponding to the updated version of the rendered representation; and

restoring the semantic representation for the requested multimedia content corresponding to the updated version of the rendered representation using the updated version of the semantic content.

21. The method for retrieving multimedia data of claim 9, wherein:

the request includes an index corresponding to the requested multimedia content; and

determining whether data corresponding to the requested multimedia content is disposed in a storage medium includes searching the storage medium using a retrieval mechanism adapted to search and retrieve content based on the index corresponding to the requested multimedia content.

22. The method for retrieving multimedia data of claim 21, wherein the retrieval mechanism includes using at least one of:

a database query,

index files, and

a file system directory structure.

23. A rendered cache comprising:

a storage medium; and

an indexing mechanism adapted to store and retrieve:

a rendered representation of the multimedia content formatted for rapid play, the multimedia content includes at least one multimedia element, the at least one multimedia element includes at least one of graphical images, audio, text, and full motion video; and

semantic content of the multimedia content, the semantic content includes data describing at least one of: characteristics of the rendered representation, and relationships between different multimedia elements disposed in the rendered representation.

24. The rendered cache of claim 23, wherein the semantic content includes data corresponding to at least one of:

a location of a hyperlink;

a size of the hyperlink;

a shape of the hyperlink;

a target index of the hyperlink;

a size of a portion of the multimedia content disposed in a rendered image;

a relative location of the portion of the multimedia content disposed in a rendered image

sizes of animated formatted graphics;

locations of animated formatted graphics;

timing of animated formatted graphics;

browser commands for addressable processing equipment; and

data corresponding to play instructions.

25. The rendered cache of claim 24, wherein the play instructions include at least one of timing of the multimedia content and ordering of the multimedia content.

26. The rendered cache of claim 23, wherein the indexing mechanism includes processing resources for converting the rendered content address into a unique index, the unique index is adapted to store and retrieve the rendered content and semantic content.

27. A client comprising:

processing resources adapted to detect a rendered representation of multimedia content and semantic content of the rendered representations, the multimedia content includes at least one multimedia element, the at least one multimedia element includes at least one of graphical images, audio, text, and full motion video, the semantic content includes data describing at least one of: characteristics of the rendered representation, and relationships between different multimedia elements disposed in the rendered representation; and

processing resources adapted to respond to detecting the rendered representation of the multimedia content and the semantic content by playing at least a portion of the rendered representation according to the semantic content.

28. The client of claim 27, wherein the semantic content includes data corresponding to hyperlinks.

29. The client of claim 27, wherein:

the client includes a set-top box;

only a portion of the multimedia content, and only a portion of the semantic content are received by the processing resources adapted to detect a rendered representation of multimedia content and semantic content of the rendered representations; and

the semantic content includes data corresponding to scroll commands.

30. A system for using multimedia content comprising:

web crawler processing resources adapted to access the multimedia content from source data storage, the multimedia content includes at least one multimedia element, the at least one multimedia element includes at least one of graphical images, audio, text, and full motion video;

rendering processing resources adapted to:

generate a semantic representation of a rendered representation of the multimedia content, the semantic representation describes at least one of: characteristics of the rendered representation of the multimedia content, and relationships between different multimedia elements disposed in the rendered representation of the multimedia content;

format the semantic representation as semantic content; and

render the multimedia content into the rendered representation, the rendered representation is formatted for rapid play; and

a rendered cache including:

a storage medium; and

an indexing mechanism adapted to store and retrieve:

the rendered representation of the multimedia content; and

the semantic content of the multimedia content;

31. The system for using multimedia content of claim 30 including:

a browser adapted to:

interpret the semantic content; and

control play of the rendered representation of the multimedia content, and

multimedia playing processing resources adapted to play the rendered representation of the multimedia content.

32. The system for using multimedia content of claim 30, wherein:

source data storage includes at least one of the Internet and a web proxy cache; and

the rendering processing resources are adapted to layout the multimedia content.

33. The system for using multimedia content of claim 31, wherein the browser processing resources:

communicate with the web crawler processing resources, the rendering processing resources, and the multimedia playing processing resources; and

are adapted to control the interactions of at least one of:

the web crawler processing resources;

the rendering processing resources, and

the multimedia playing processing resources.

34. A system for accessing multimedia content comprising:

a rendered cache including:

a storage medium; and

an indexing mechanism adapted to store and retrieve:

a rendered representation of the multimedia content formatted for rapid play, the multimedia content includes at least one multimedia element, the at least one multimedia element includes at least one of graphical images, audio, text, and full motion video; and semantic content of the multimedia content, the semantic content includes data describing at least one of: characteristics of the rendered representation, and relationships between different multimedia elements disposed in the rendered representation; and

rendering processing resources adapted to:

convert the multimedia content into the rendered representation, the rendered representation is formatted for rapid play; and

create a graphical representation of the multimedia content.

35. The system for accessing multimedia content of claim 34 including: multimedia play processing resources adapted to:

read the rendered representation; and

play the multimedia content on a display; and

browser processing resources adapted to interpret the semantic content.

36. The system for accessing multimedia content of claim 34 including:

layout processing resources adapted to:

derive rendering instructions from a content definition;

lay out the multimedia content; and

generate a semantic representation of the multimedia content from lay out of the multimedia content; and

wherein the rendering processing resources use the rendering instructions to create the graphical representation.

37. The system for accessing multimedia content of claim 35, wherein the browser processing resources are adapted to control play of the multimedia content.

38. A method for playing multimedia content, comprising:

retrieving a rendered representation of the multimedia content from a storage medium;

retrieving semantic content of the rendered representation from the storage medium, the semantic content including data describing at least one of: characteristics of the rendered representation, and relationships between different multimedia elements disposed in the rendered representation;

browser processing resources reading the rendered representation and the semantic content;

the browser processing resources restoring a semantic representation based on the semantic content;

the browser processing resources transmitting:

an active portion of the rendered representation, the active portion of the rendered representation is one of: a portion of the rendered representation presently being played, and a portion of the rendered representation to be played rapidly after transmitting; and

an active portion of the semantic content corresponding to the active portion of the rendered representation;

client processing resources detecting the active portion of the rendered representation and the active portion of the semantic content; and

the client processing resources playing the active portion of the rendered representation.

39. The method for playing multimedia content of claim 38, wherein:

the client includes a set-top box;

the multimedia content includes an image having at least one hyperlink;

the semantic representation including at least one of:

a location of at least one hyperlink;

a size of at least one hyperlink;

a shape of at least one hyperlink; and

a target index of at least one hyperlink.

40. An apparatus for storing multimedia data, comprising:

means for detecting multimedia content including layout instructions, the multimedia content includes at least one multimedia element, the at least one multimedia element includes at least one of graphical images, audio, text, and full motion video;

means for laying out the multimedia content according to the layout instructions to form rendering instructions and a semantic representation of a rendered representation of the multimedia content, the semantic representation describes at least one of: characteristics of the rendered representation of the multimedia content, and relationships between different multimedia elements disposed in the rendered representation of the multimedia content;

means for rendering the multimedia content according to the rendering instructions to produce the rendered representation;

means for storing the rendered representation; and

means for storing data corresponding to the semantic.

41. A rendered cache comprising:

means for storing data corresponding to multimedia content, the data including:

a rendered representation of the multimedia content formatted for rapid play, the multimedia content includes at least one multimedia element, the at least one multimedia element includes at least one of graphical images, audio, text, and full motion video; and

semantic content of the multimedia content, the semantic content includes data describing at least one of: characteristics of the rendered representation, and relationships between different multimedia elements disposed in the rendered representation

indexing means for storing and retrieving:

the rendered representation; and

the semantic content.

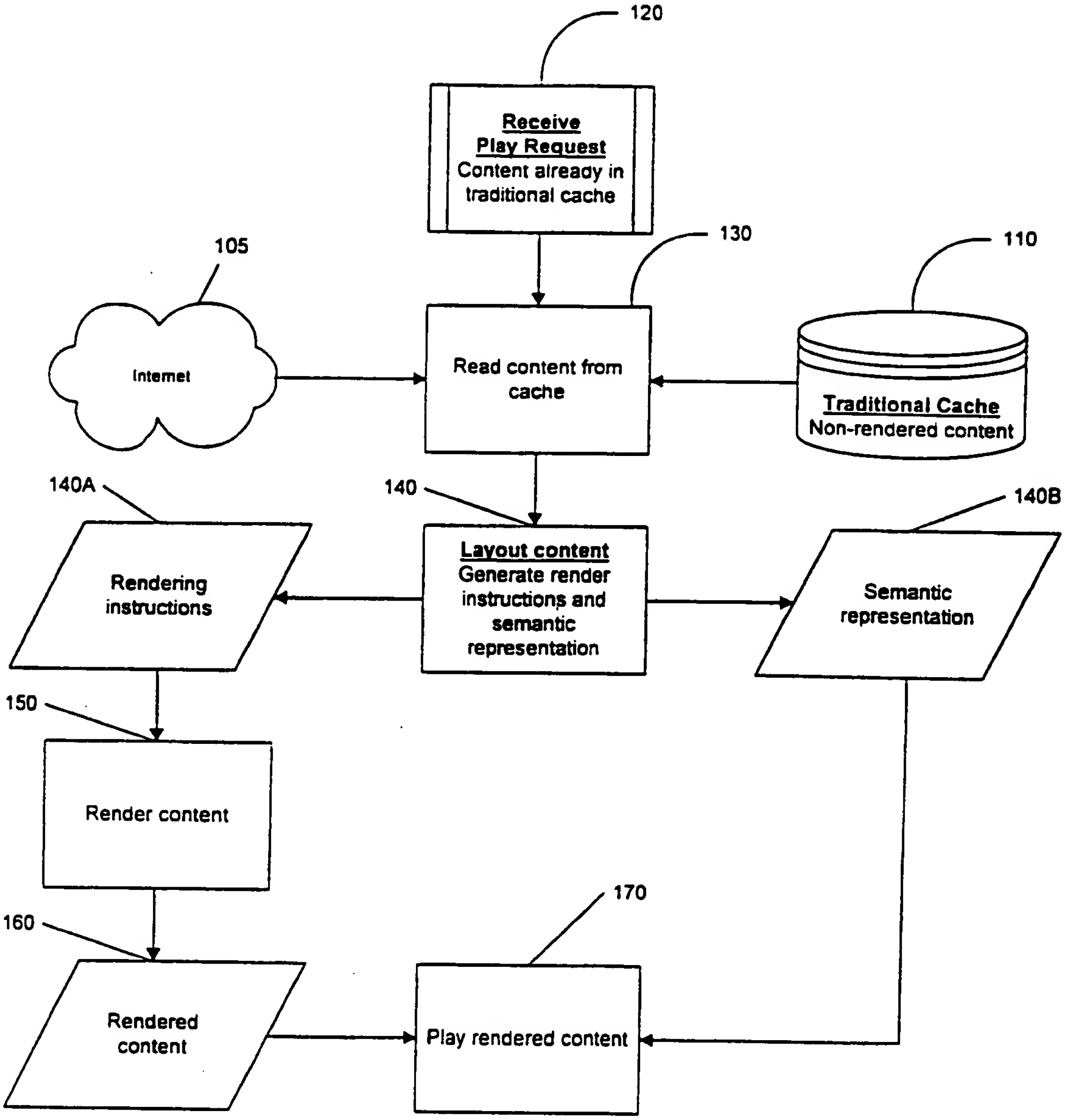


Fig. 1: Prior Art Process Details

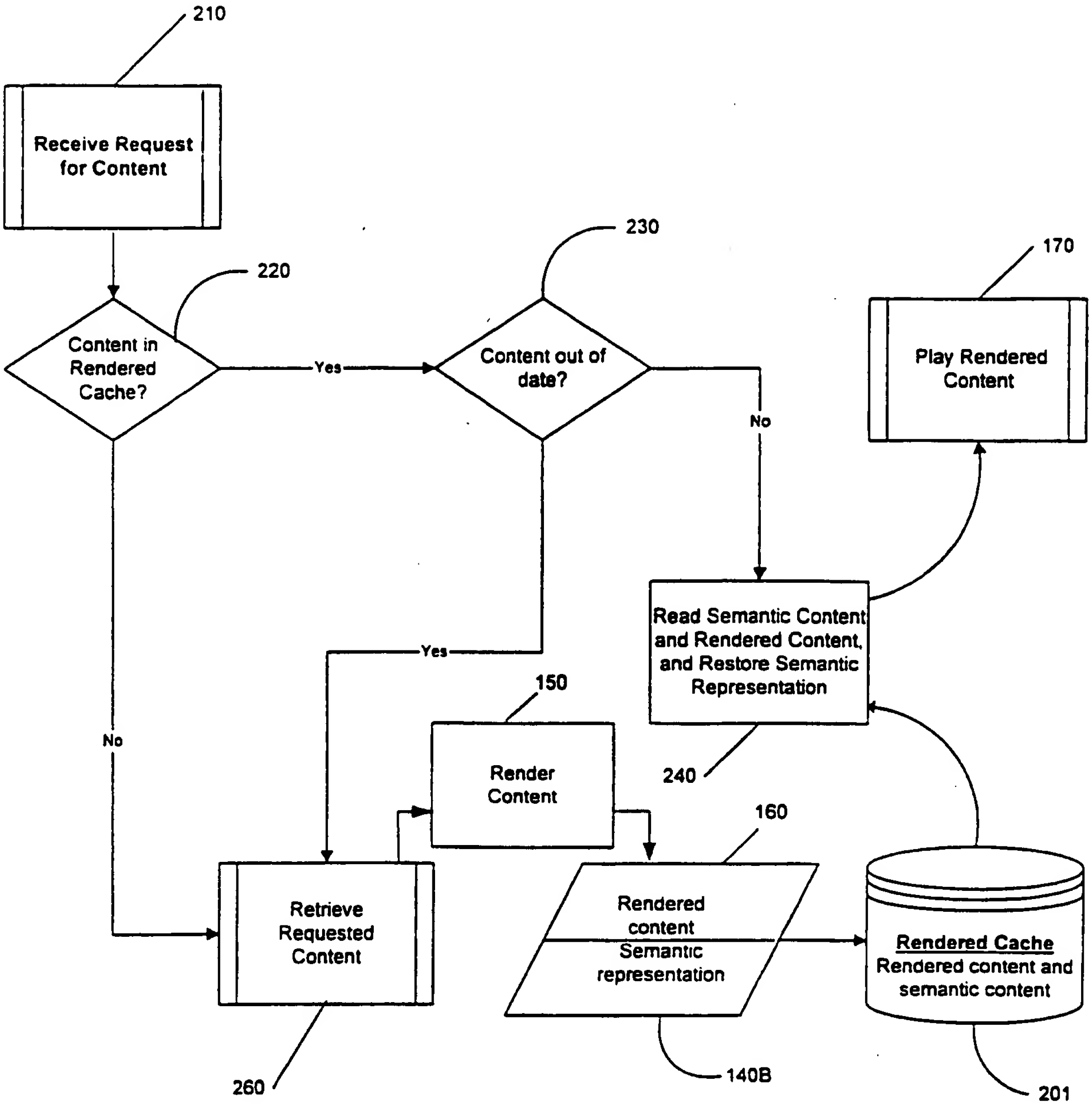

210
Receive Request for Content
220
Content in Rendered Cache?
Yes
230
Content out of date?
No
170
Play Rendered Content
Read Semantic Content and Rendered Content, and Restore Semantic Representation
240
Yes
No
150
Render Content
160
Retrieve Requested Content
260
Rendered content
Semantic representation
140B
Rendered Cache
Rendered content and semantic content
201

Fig. 2: Rendered Cache Process Overview

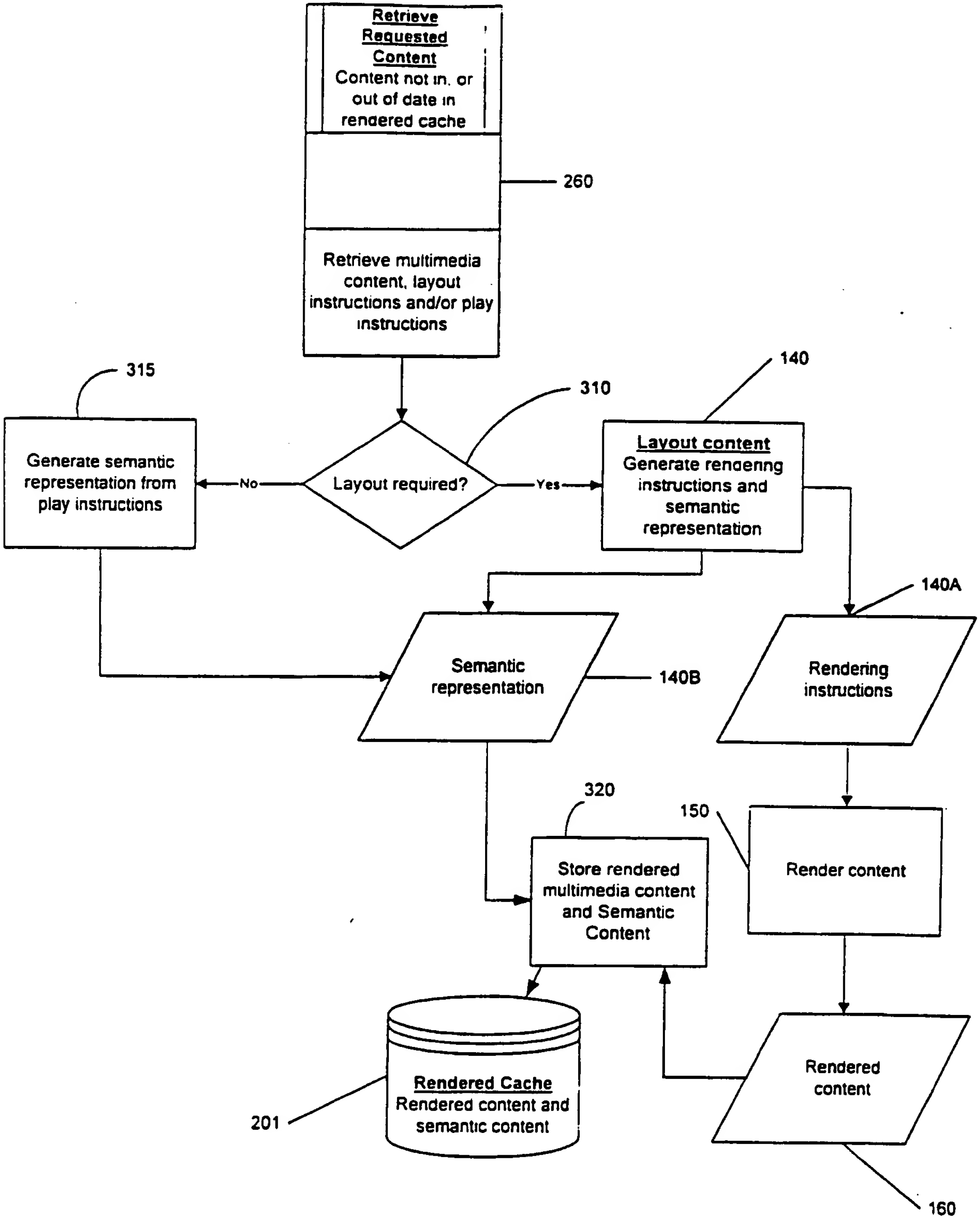

Retrieve Requested Content
Content not in. or out of date in rendered cache
260
Retrieve multimedia content, layout instructions and/or play instructions
315
310
140
Generate semantic representation from play instructions
No
Layout required?
Yes
Layout content
Generate renaering instructions and semantic representation
140A
Semantic representation
140B
Rendering instructions
320
150
Store rendered multimedia content and Semantic Content
Render content
Rendered Cache
Rendered content and semantic content
201
Rendered content
160

Fig. 3: Render Process Details

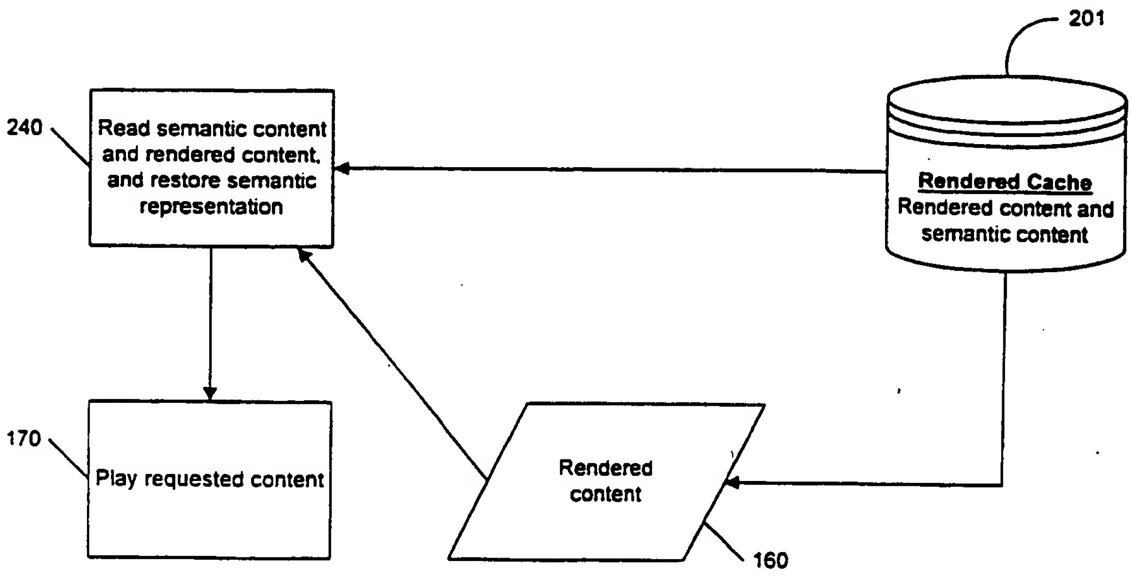

201
240
Read semantic content and rendered content, and restore semantic representation
Rendered Cache
Rendered content and semantic content
170
Play requested content
Rendered content
160

Fig. 4: Play Process Details

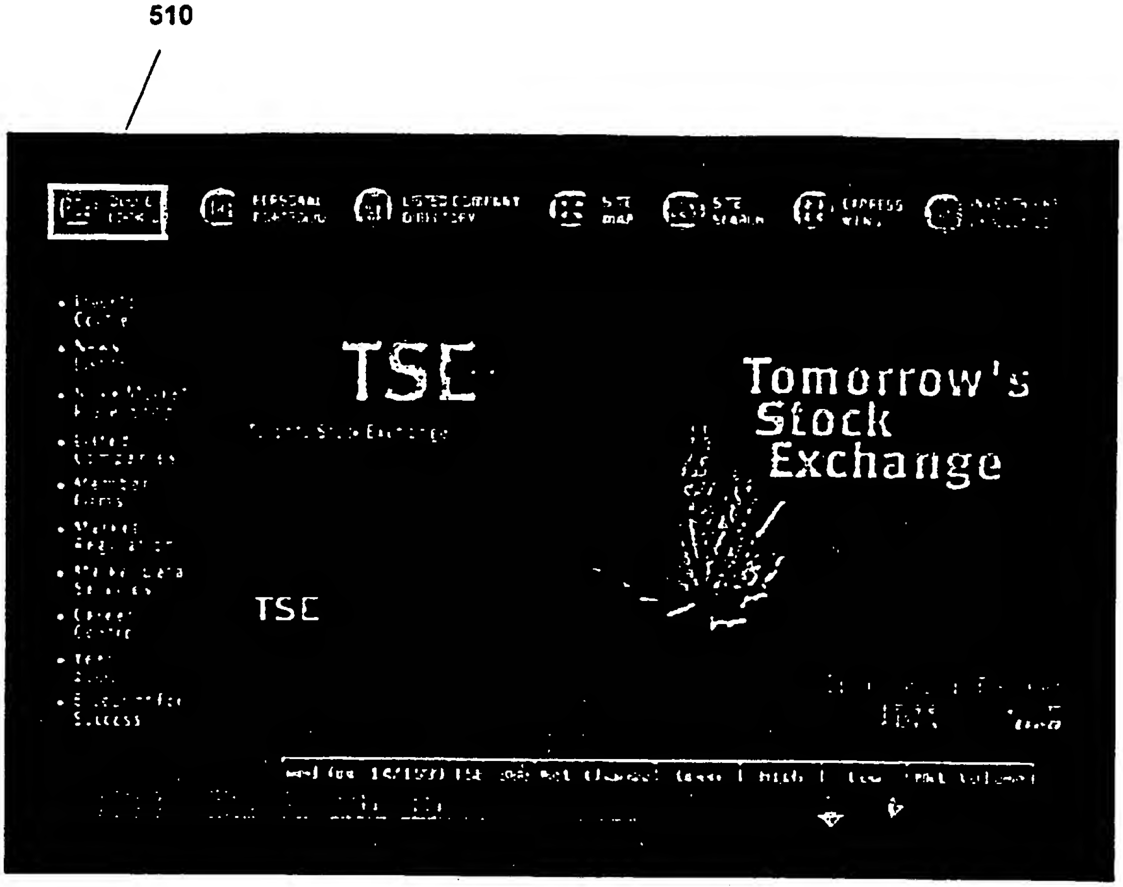

510
PERSONAL PORTFOLIO
LISTED COMPANY DIRECTORY
SITE MAP
SITE SEARCH
Listed Companies
Member Firms
Market Regulation
Career Centre
Success
TSE
TSE
Tomorrow's Stock Exchange
High
Low

FIGURE 5A

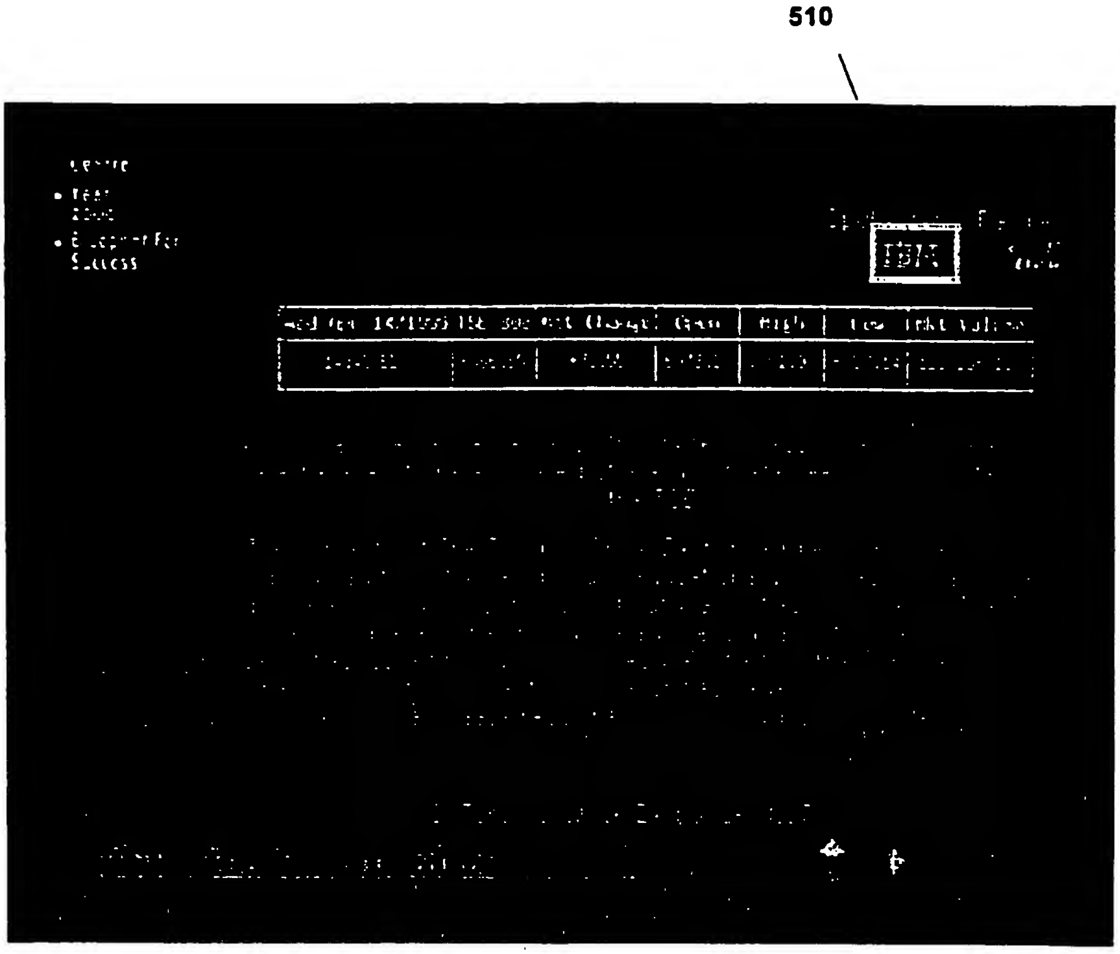


FIGURE 5B

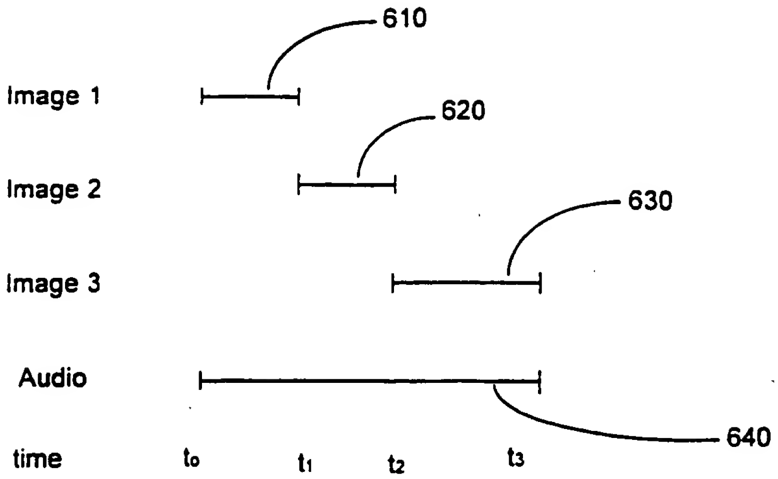


Fig 6A: Presentation timeline

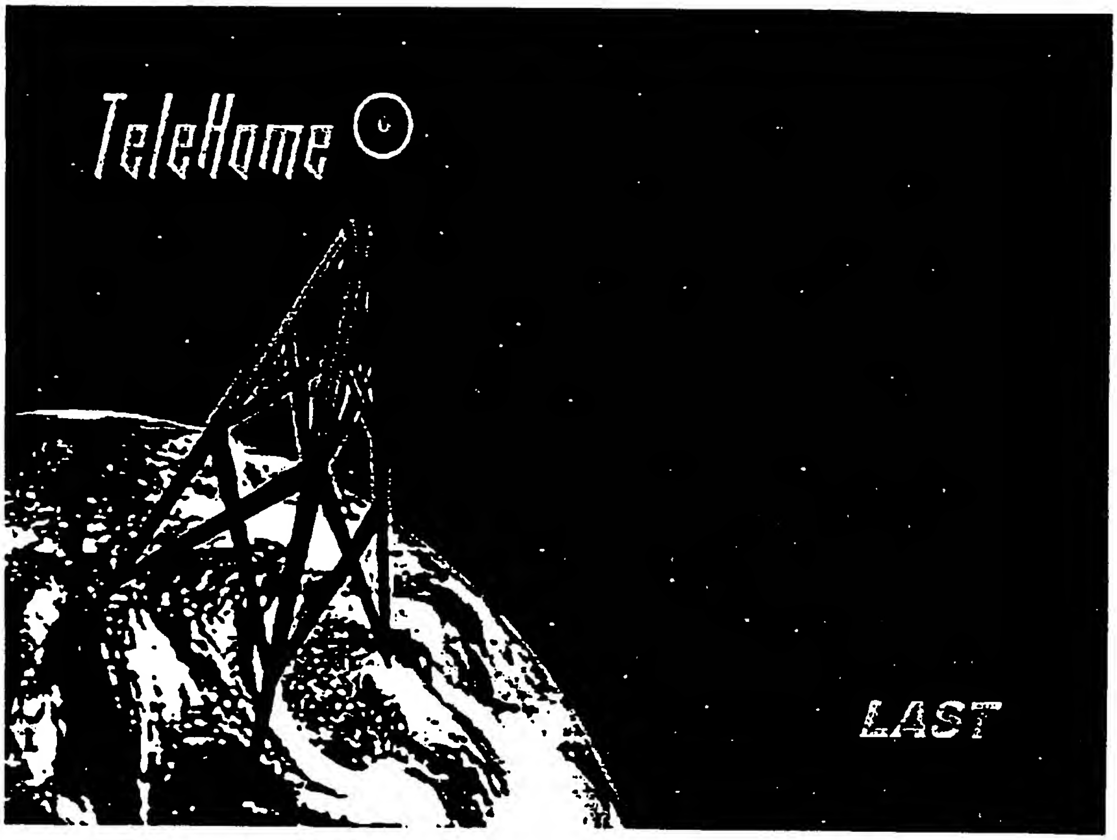


FIGURE 6B

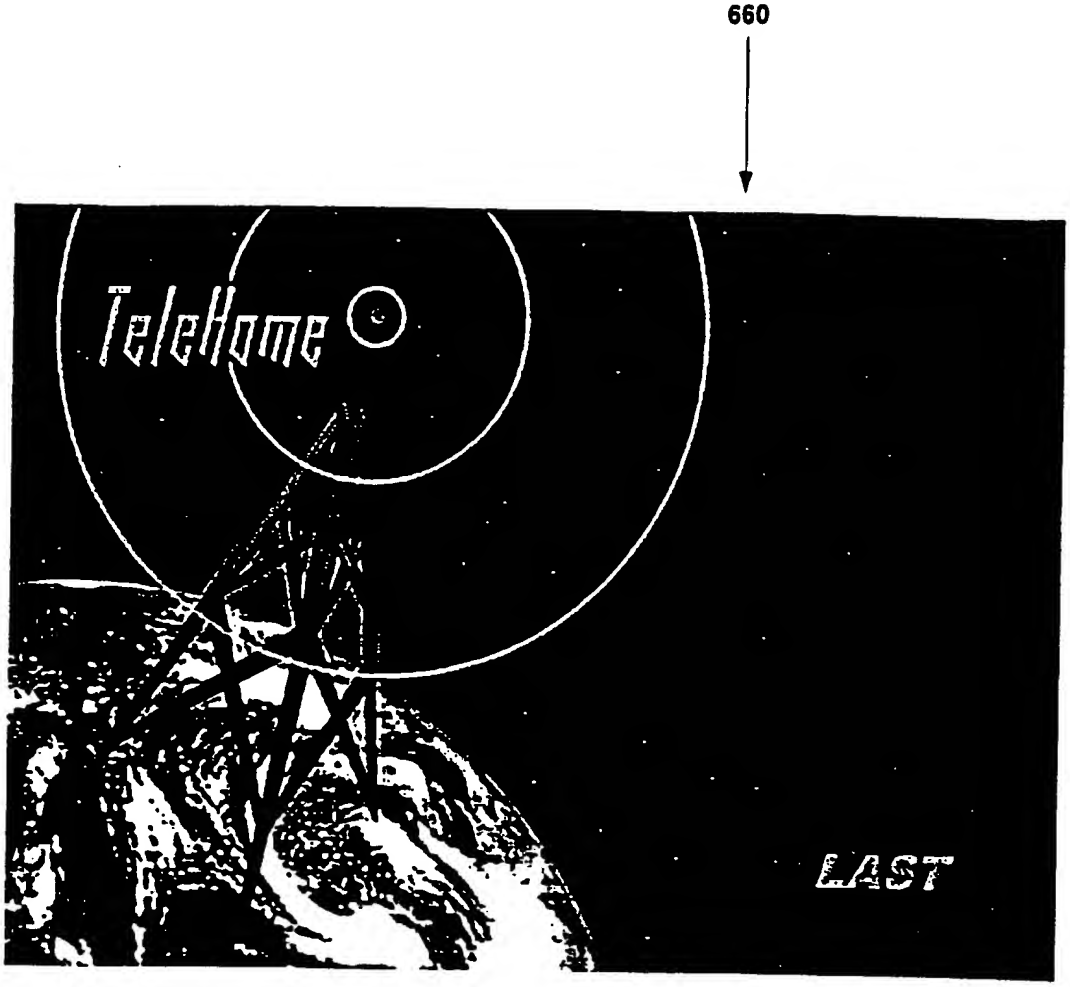


FIGURE 6C

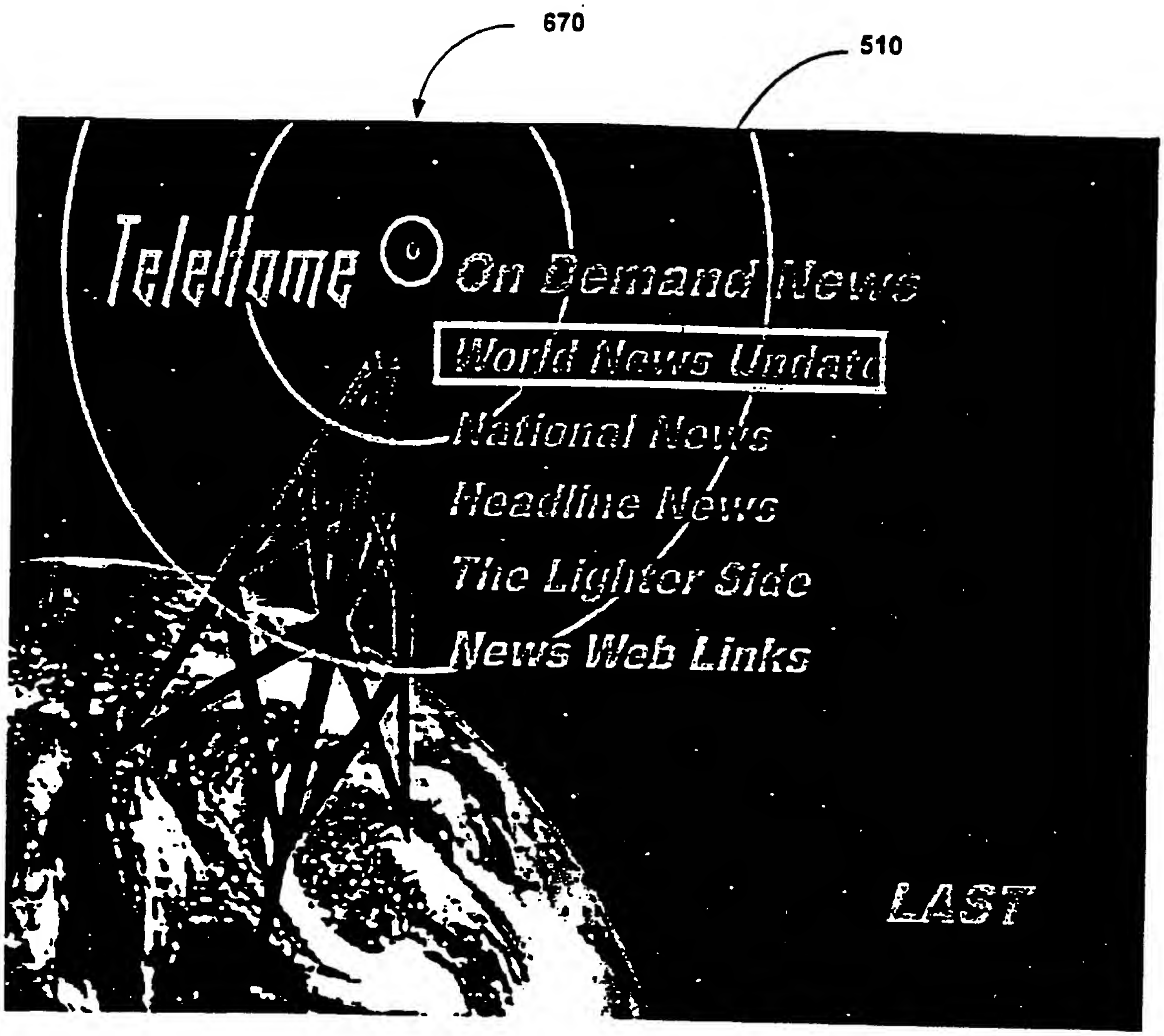

670
510
TeleHome
On Demand News
World News Updat
National News
Headline News
The Lighter Side
News Web Links
LAST

FIGURE 6D

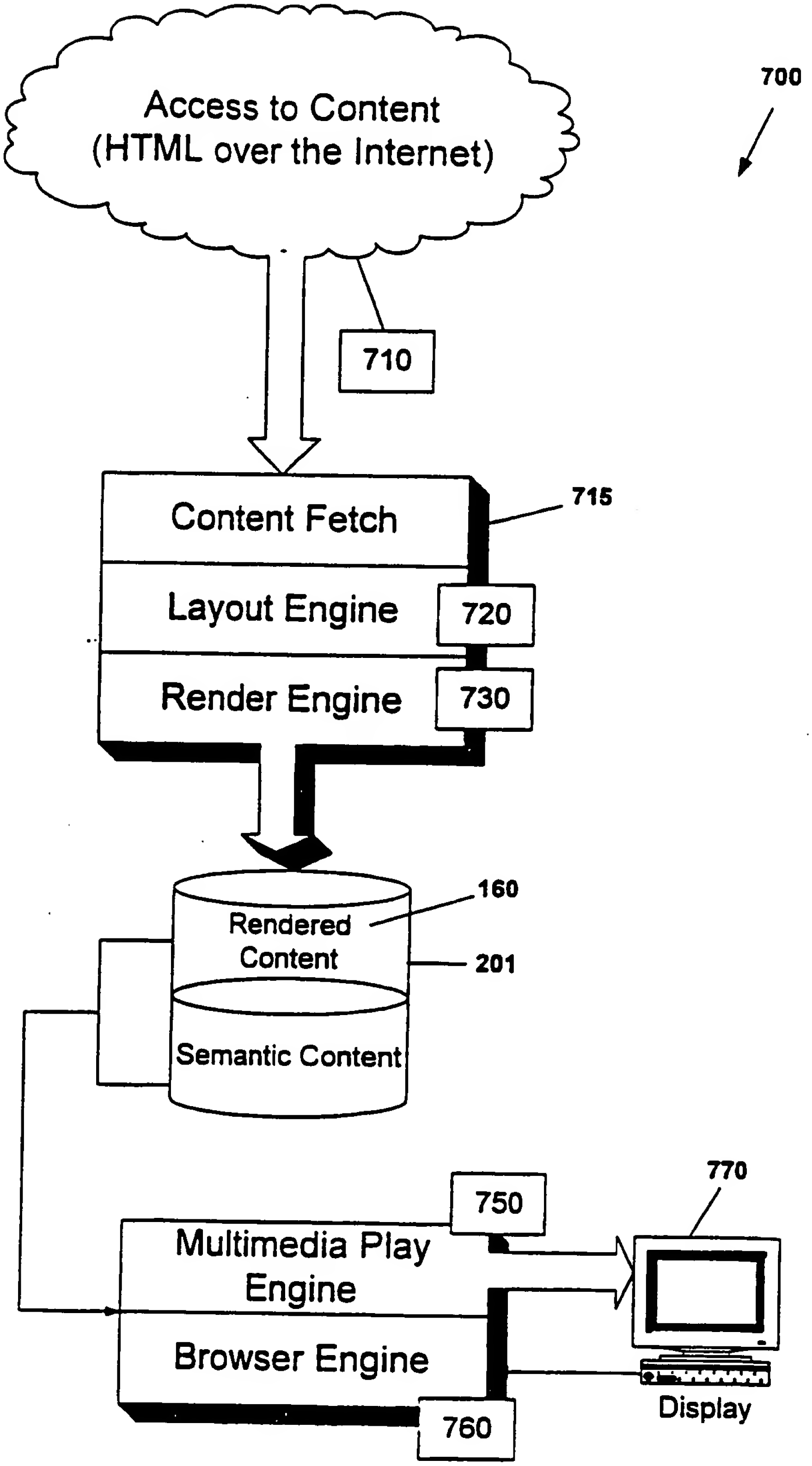

Access to Content
(HTML over the Internet)
700
710
Content Fetch
715
Layout Engine
720
Render Engine
730
Rendered Content
160
201
Semantic Content
750
770
Multimedia Play Engine
Browser Engine
760
Display

FIGURE 7